Hybrid Method Simulation of Slender Marine Structures

Christiansen, Niels Hørbye

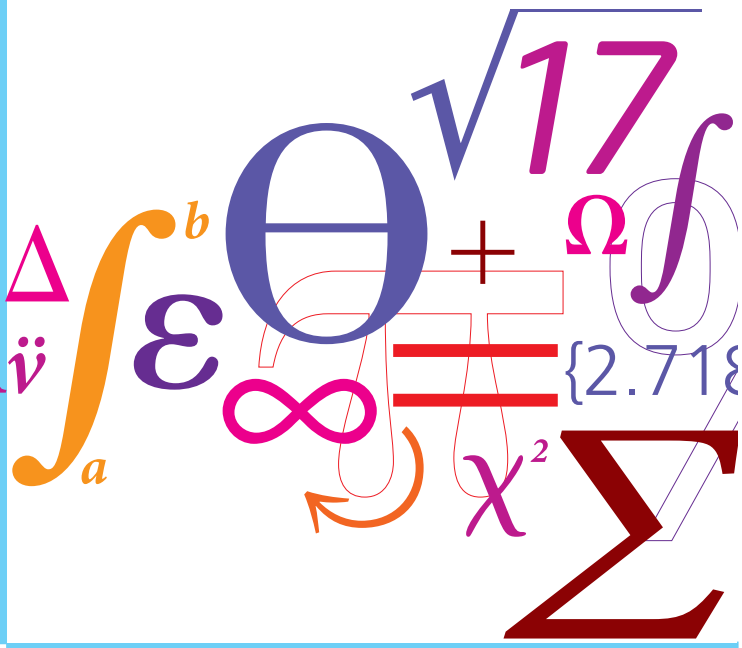
Publication date:
2014

Document Version
Publisher's PDF, also known as Version of record

Hybrid Method Simulation of Slender Marine Structures

PhD Thesis

$$(EIv'')''=q-\rho A\ddot{v}$$



Niels Hørbye Christiansen
DCAMM Special Report No. S182
August 2014

DTU Mechanical Engineering
Department of Mechanical Engineering

Hybrid Method Simulation of Slender Marine Structures

Niels Hørbye Christiansen

August 2014

Department of Mechanical Engineering
Section of Solid Mechanics
Technical University of Denmark

Published in Denmark by
Technical University of Denmark

Solid Mechanics
Department of Mechanical Engineering
Technical University of Denmark
Nils Koppels Alle, Building 404, DK-2800 Kgs. Lyngby, Denmark
Phone +45 4525 1960, Telefax +45 4593 1475
E-mail: `info@mek.dtu.dk`
WWW: `http://www.mek.dtu.dk/`

Publication Reference Data
Christiansen, N. H.
Hybrid Method Simulation of Slender Marine Structures
PhD Thesis
Technical University of Denmark,
Section of Solid Mechanics.
August, 2014
ISBN 978-87-7475-410-7
DCAMM Special Report No.: S182

Preface

The work leading to this thesis has been conducted between the spring of 2010 and the spring of 2014. The project work has been carried out in a cooperation between DNV GL and the Technical University of Denmark (DTU) and has been partly funded by DNV GL and partly by the Danish Ministry of Science, Innovation and Higher Education through the Industrial PhD Programme.

Besides the necessary financial funding, I have also been fortunate to receive invaluable personal support throughout the process, without which the realization of this project would never have been possible. I would like to express my deepest gratitude to my two supervisors. Nils Sødahl for accepting the role as company supervisor, for providing contact to relevant people and for patiently reading through papers, applications, budgets etc. and giving relevant and constructive feedback. Thanks to my university supervisor Jan Høgsberg for his never wavering support and commitment. The guidance has been indispensable and hugely appreciated.

I am in great debt to Per Erlend Voie from the DNV GL office in Trondheim, Norway, for his tremendous work with setting up numerical models and generating data for me to use in my work. Without his assistance, the developed method would not have been tested on numerical models of realistic structures.

As I did not have any prior knowledge about machine learning techniques I had to attend a number of courses on that subject during the initial stages of my studies. During these courses I was lucky enough to meet Ole Winther - associate professor at DTU Compute. Ole has subsequently been a great help for me as coauthor on a number of papers.

I owe thanks to my line manager Søren Esdahl who were not involved in the formulation and setup of this project. Nevertheless, even though my only contribution to his department throughout the last couple of years have been to figure negatively in his account, he has supported me in the best possible fashion all the way.

I truly hope that the future will bring a lot of interesting cooperation with all the people mentioned above.

Finally I would like to thank my family for their endless support and patience with me.

Søborg, Denmark, 2014

Niels Hørbye Christiansen

Resumé

Denne afhandling omfatter en indledende sammenfatning og fem uddybende artikler, som alle omhandler forskellige aspekter af implementering af en hybridmetode, som kombinerer klassiske simuleringsmetoder med neurale netværk. Afhandlingen dækker tre hovedområder. Fælles for alle områder er, at de beskæftiger sig med tidsdomæne simuleringer af slanke marine konstruktioner - dvs. forankringsliner eller fleksible olierør til anvendelse i forbindelse med offshore-installationer på dybt vand.

Den første del af afhandlingen beskriver hvordan neurale netværk kan designes og trænes til, at dække et stort antal forskellige bølgetilstande. Neurale netværk er indrettet således, at de kun kan genkende mønstre, som er indeholdt i de data, der benyttes til træningen af netværket. Udmattelsesanalyser af marine konstruktioner indbefatter ofte simuleringer med mere end hundrede forskellige bølgetilstande. Dette betyder, at skal metoden være rigtig effektiv, må træningsdata arrangeres således, at ét neuralt netværk kan dække alle relevante bølgetilstande. Metoden demonstreres på en numerisk model af en forankringsline til en flydende platform.

Den næste del af afhandlingen undersøger hvordan serieforbundne neurale netværk kan benyttes til at simulere dynamisk respons af alle kritiske områder på fleksible rør. Ved design af forankringsliner kigger man udelukkende på snitkræfter i toppen af linerne. Disse kræfter er relativt simple at bestemme ved brug af et enkelt neuralt netværk. Ved rørdesign skal man, afhængigt af hvilken rørkonfiguration det drejer sig om, undersøge en række kritiske områder. I denne forbindelse er der ikke nødvendigvis en direkte sammenhæng mellem ydre belastninger og resulterende snitkræfter, hvilket betyder at et enkelt neuralt netværk ikke altid er tilstrækkeligt. Det demonstreres hvordan dette problem kan løses ved at opsætte en serie af neurale netværk som trinvist simulerer sig vej igennem konstruktionen.

Endeligt omhandler den sidste del af afhandlingen optimering af neurale netværk. Det gives eksempler på hvordan trænede neurale netværk kan reduceres voldsomt i størrelse og stadig opretholde en høj simuleringsnøjagtighed. Udover at opnå et mere kompakt netværk, kan denne optimeringsprocedure benyttes til at rangere vigtigheden af de ydre belastninger på konstruktioner. Sådanne følsomhedsstudier kræver normalt et stort antal simuleringer, men med denne metode, kan de gennemføres ved brug af kun et enkelt sæt data.

Den store fordel ved hybridmetoden er, at den giver anledning til en markant besparelse i beregningstid i forbindelse med ikke-lineære dynamiske tidsdomæne simuleringer. Da de neurale netværk ikke kan stå alene, men er afhængige af modeller der genererer træningsdata, må man hele tiden vurdere om tidsbesparelsen ved hybridmetoden opvejes af det ekstra arbejde forbundet med at etablering af metoden.

Abstract

This present thesis consists of an extended summary and five appended papers concerning various aspects of the implementation of a hybrid method which combines classical simulation methods and artificial neural networks. The thesis covers three main topics. Common for all these topics is that they deal with time domain simulation of slender marine structures such as mooring lines and flexible risers used in deep sea offshore installations.

The first part of the thesis describes how neural networks can be designed and trained to cover a large number of different sea states. Neural networks can only recognize patterns similar to those comprised in the data used to train the network. Fatigue life evaluation of marine structures often considers simulations of more than a hundred different sea states. Hence, in order for this method to be useful, the training data must be arranged so that a single neural network can cover all relevant sea states. The applicability and performance of the present hybrid method is demonstrated on a numerical model of a mooring line attached to a floating offshore platform.

The second part of the thesis demonstrates how sequential neural networks can be used to simulate dynamic response of specific critical hot spots on a flexible riser. In the design of mooring lines only top tension forces are considered. These forces can easily be determined by a single neural network. Riser design, depending on the applied configuration, requires detailed analysis of several critical hot spots along the structure. This means that the relation between external loading and corresponding structural response not necessarily is as direct as for the mooring line example. Hence, one neural network is not sufficient to cover the entire structure. It is demonstrated how a series of neural networks can be set up to sequentially simulate the dynamic response at critical locations along a complex riser structure.

The final part of the thesis deals with the optimization of neural networks. It is shown how trained networks can be dramatically reduced in size while still maintaining a high simulation accuracy. Beside providing a more compact neural network the optimization procedures can be used to rank the importance of external effects on structures. Such sensitivity studies usually require numerous simulations. But by using this method these studies can be based on just one short simulation sequence which reduces the computational cost significantly.

The great advantage with the hybrid method is that it gives rise to significant reductions in computation time associated with nonlinear dynamic time domain simulations. However, since the neural network depends on pre-generated training data, one must always consider the balance between saved computation time and time spend on establishing the hybrid method.

Publications

Appended papers:

[P1] N. H. Christiansen, P. E. T. Voie, J. Høgsberg, N. Sødahl
Efficient mooring line fatigue analysis using a hybrid method time domain simulation scheme
Proceedings of the OMAE 2013 International Conference on Ocean, Offshore and Arctic Engineering

[P2] N. H. Christiansen, P. E. T. Voie, O. Winther, J. Høgsberg
Comparison of neural network error measures for simulation of slender marine structures
Journal of Applied Mathematics, (2014)

[P3] N. H. Christiansen, P. E. T. Voie, O. Winther, J. Høgsberg
Optimization of Neural Networks for Simulation of Slender Marine Structures
Institution of Mechanical Engineers. Proceedings. Part M: Journal of Engineering for the Maritime Environment, (2015)

[P4] N. H. Christiansen, P. E. T. Voie, J. Høgsberg, N. Sødahl
Optimized mooring line simulation using a hybrid method time domain scheme
Proceedings of the OMAE 2014 International Conference on Ocean, Offshore and Arctic Engineering

[P5] N. H. Christiansen, J. Høgsberg
Sequential Neural Network Procedure for Nonlinear Time Domain Simulation of Offshore Risers
Submitted August 2014

Additional contributions:

N. H. Christiansen, J. Høgsberg, O. Winther
Artificial Neural Networks for Nonlinear Dynamic Response Simulation in Mechanical Systems
Proceedings of the 24th Nordic Seminar on Computational Mechanics
J. Freund and R. Kouhia (Eds.), Helsinki, Finland, November 2011.

N. H. Christiansen, J. H. Job, K. Klyver, J. B. Høgsberg
Optimal Brain Surgeon on Artificial Neural Networks in Nonlinear Structural Dynamics
Proceedings of the 25th Nordic Seminar on Computational Mechanics
K. Persson, J. Revstedt, G. Sandberg, M. Wallin (Eds.), Lund, Sweden, November 2012

N. H. Christiansen
How childplay inspired new artificial brain
www.sciencenordic.com
Winner of the 2nd prize at the Danish Industrial PhD Association - 2013 Communication Prize competition.

N. H. Christiansen
Menneskets hjerne afslører bygningers svagheder
www.videnskab.dk
Winner of the 3rd prize at the Danish Industrial PhD Association - 2014 Communication Prize competition.

Contents

1 Introduction

The overall objective with this project has been to develop a method that can reduce the excessive computational effort associated with nonlinear dynamic analysis of flexible marine structures such as mooring lines and risers. In design and analysis of slender risers and mooring systems for floating offshore units, there is a very pronounced need for long term time series simulations - both in evaluation of ultimate limit state and of long term fatigue [1, 2, 3, 4]. Slender offshore structures exhibit large deformations and therefore require non-linear models for reliable analysis. The combination of the need for detailed nonlinear models and the demand for long time series simulations makes design and analysis of these types of structures very time consuming and costly. And even though numerous suggestions on how to reduce the amount of required simulation data have been suggested, se e.g. [5, 6, 7, 8], this computational task remains an inevitable part of the analysis of this type of structures.

The use of artificial neural networks (ANNs) for reducing computational cost has shown to be useful in various aspects of structural engineering. Adeli [9] gives an overview of applied methods, while Waszczyszyn & Zieminański [10] demonstrate different possible applications of neural networks in mechanical engineering. Many of these applications deal with detection of structural fatigue damage. Gupta [11] applies artificial neural networks and principal component analysis in detection of fatigue damage by using the two tools as outlier detectors on observed data sequences. Javier et al. [12] uses a similar approach to detect damages in bridge-like structures. Also within nonlinear structural response analysis artificial neural networks have shown to be a powerful tool. Most of the techniques are based on hybrid methods combining finite element models and ANNs. Ordaz-Hernandez et. al. [13] demonstrate how ANNs can be trained to predict the deflection of a nonlinear cantilevered beam. In dynamic analysis Xu et. al. [14] identifies structural parameters of a five story building using a neural network on a pre-generated dynamic time domain response history. Guarize et al. [15] applied a similar network structure to simulate the dynamic response of a flexible risers in service, thereby reducing calculation time by a factor of about 20. The ability of the ANN to perform nonlinear mapping between a given input and a system output makes it capable of performing response predictions without time consuming equilibrium iterations. Hence, a properly trained ANN can save a lot of time in response simulations and thereby help to avoid the infuriating compromise between model sophistication and computational efficiency.

Time domain dynamic analysis of structures is in practise a matter of solving a system of differential equations. Since these equations very rarely offer an explicit solution method for numerical integration must be applied [16, 17]. Even with efficient algorithms implemented in modern high performance computers, solving nonlinear differential equations remain a very time consuming task. The idea with this project is, instead of solving these equations, to simulate the solution to the equations by use of artificial neural networks. The artificial neural network is a mathematical tool inspired by the network-like structure

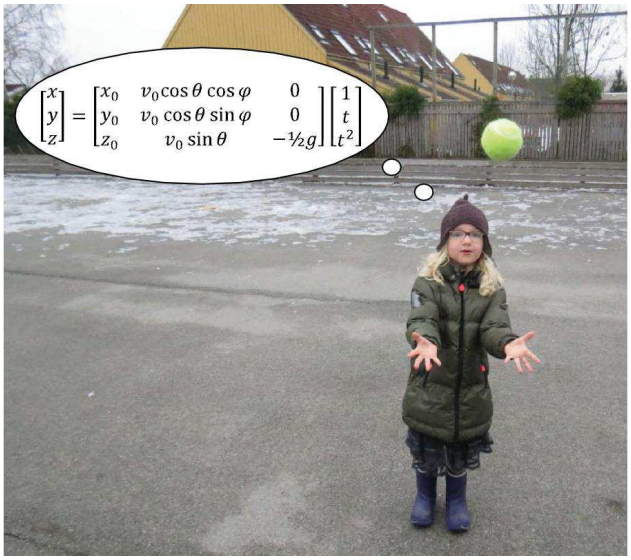


Figure 1: The brain constantly simulates solutions to equations of motion.

of the human brain [18, 19, 20]. Due to its structure the human brain has a remarkable ability to recognize patterns with an impressive speed. It is this ability that makes us able to read hand written letters and to recognize faces. This ability to recognize patterns can also be utilized to predict patterns of movement. An example of this is when a girl catches a flying tennis ball as in Figure 1. When deciding where to put her hands she obviously does not think about the set of equations that describe the trajectory of the ball through the air. Instead, the girl recognizes the situation and is able to put her hands at the right place at the right time. This process provides a very fast and accurate simulated solution to the equations that describe the underlying laws of nature which govern the trajectory of the ball. Hence, since the girl has seen many balls flying through the air she is capable of recognizing and predicting the trajectory of the ball and thereby to bypass the need for an actual solution to the governing equations of motion.

To replicate this ability mathematically requires two things. 1: A mathematical formulation with a brain-like network structure such as the ANN and 2: a set of training data to train the network to recognize a given pattern. The training data corresponds to letting the girl see a series of tennis balls fly through the air. A girl who has never seen a flying ball will most likely be unable to catch it. However, after relatively few attempts the girl will recognize a pattern in the way a ball falls and also be able to predict this pattern and hence, be able to catch the ball. For the ANN to be able to identify underlying patterns in a set of data there must be a sufficient amount data to represent the characteristics of the data generator.

It is difficult - if not impossible - to explain exactly how we recognize a certain pattern e.g. the face of an old friend. Nevertheless, the human brain is very fast and reliable when it comes to pattern recognition. The same qualities apply for the ANN. The structure of the ANN makes it impossible to extract any physical interpretation of the elements in the network as opposed to numerical models where all individual elements represent a physical property. As Warner and Misra [21] put it: "A neural network never reveals the functional relations; they are buried in the summing of the sigmoidal functions."

When it comes to complex flexible structures only very few types of analysis can be carried out satisfactorily by linear methods. This means that the vast majority of the analyzes associated with design of slender marine structures require computational demanding and time consuming nonlinear analysis. Numerical analysis of structures is always a balance between model sophistication and computational effort. Often linear analysis is not adequate to obtain satisfactorily accurate results. Over the last decades various methods for applying pattern recognition tools in order to save computational effort have been suggested. As mentioned, Guarize et. al [15] showed that the hybrid method, if only for very distinct types of analysis, can introduce dramatic reductions in computation time in connection with dynamic time domain simulations of riser systems. The bold ambition of this PhD project has been to develop and expand this idea into a set of useful tools ready for implementation. The different analysis regimes are illustrated in Figure 2 with respect to computational cost and ability to comprise structural complexity. Nonlinear numerical models can include a high degree of structural complexity but are often computationally demanding. Linear numerical models, on the other hand, serve to limit the computational effort but do that by sacrificing model complexity. The advantage of a well trained ANN is that it can include nonlinear features while matching the speed of linear methods.

Beside being able to perform nonlinear mapping the ANN comprise another interesting feature. That is the robustness of the network structure. If just one calculation step is deleted in a regular computer the whole process breaks down. If the functionality of the human brain also collapsed if just one neural connection was damaged we would be in serious trouble. One of the big advantages with the structure of the brain is its robustness towards minor damages.

Actually our brain frequently suffers from minor damages in the neural network without impairing the overall performance of the brain. This robustness is due to the parallel network structure. In case of severe brain damage a person can loose certain abilities completely. However, through rehabilitation it is often possible to regain the lost abilities. This is also due to the parallel network structure that makes it possible to let information flow around a damaged area and thereby compensate for the damage [22, 23, 24, 25]. The ability to tolerate damages also applies for artificial neural networks. This has inspired people to develop methods to conduct deliberate network pruning in order to optimize the network in terms of size and performance [26, 27, 28].

Over the course of this project the focus has been on three main extensions to the hybrid methodology. First it has been shown in the appended paper [P1]

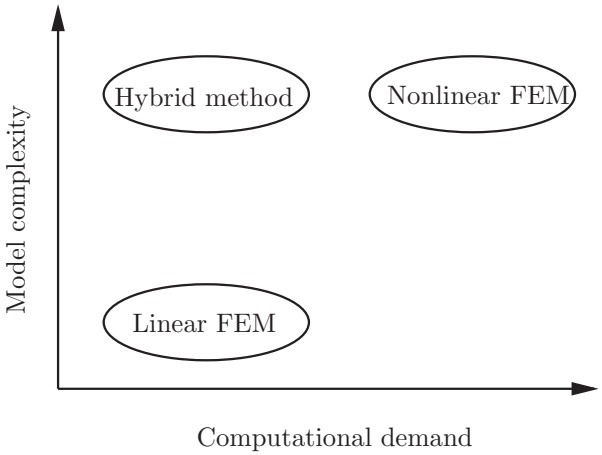


Figure 2: Hybrid method simulation can bypass the compromise between model complexity and computational effort.

that the method can be expanded to cover a complete scatter diagram with about 150 different sea states for simulation of top tension forces in a mooring line on a floating offshore platform. Secondly a dynamic response simulation of a entire flexible riser using a series of simple ANNs has been presented in [P5]. And finally, by using an optimization algorithm and adjusted error measures evaluated in [P2], it has been demonstrated in [P3,P4] that the hybrid method can be used as a very reliable tool for efficient parametric studies based on a single short simulation sequence only.

2 Artificial Neural Networks

The Artificial Neural Network is a pattern recognition tool inspired by the structure of the human brain. The ANN can be used both for classification and for regression. In this thesis only regression is considered. In most regression methods a functional form is imposed on the data. The power of the ANN approach lies in the fact that it does not assume any functional relationship, but instead the data define the functional form automatically by itself [21].

The original idea behind Artificial Neural Networks was introduced in 1943 when McCulloch and Pitts [29] presented a computational model for neural networks. In 1958 Rosenblatt created the perceptron which is a layer based algorithm for pattern recognition [30]. However, the ANN research stagnated quite abruptly in 1960's when Minsky and Papert [31] discovered two fundamental problems with the computational neural networks. The first problem was that computers of that time were not sophisticated enough to handle the computations associated with large neural networks. This problem is no longer

an issue. The second problem was that the ANN was incapable of processing the exclusive-or circuit [32]. The latter problem was solved in 1975 when Werbos [33] developed the backpropagation algorithm. Over the last decades neural networks have proved to be useful in various practical applications within the fields of finance, marketing, health, medicine, engineering etc. [34, 35, 36].

There exists vast amount of literature concerning Artificial Neural Networks and their various features and applications. A quick introduction is given by Warner and Misra in [21]. A more thorough description of the topic can be found in [37] by Bishop. All ANN related software developed in this project is based on the basic ANN toolbox code provided in connection with the course 02457 Non-Linear Signal Processing at Technical University of Denmark.

2.1 Setting up the ANN

The architecture of the ANN used throughout this thesis is shown in Figure 3. The ANN consists of an input layer, a hidden layer and an output layer. All layers consist of an arbitrary number of neurons. The size of the input and output layers depend on the specific problem to which the ANN is applied. As the neurons in the hidden layer have no physical interpretation, the size of this layer is adjustable. However, since the ability of the ANN to learn complex patterns is related to the size of this layer, it must have sufficient size to comprise all underlaying characteristics of the data in question. On the other hand, it has been shown that for a fixed amount of training data networks with too many elements do not generalize well [27]. Two neurons in neighboring layers are connected and each of these connections have a weight. The connections are represented by the straight lines in Figure 3. The training of an ANN corresponds to an optimization of these weights with respect to a particular set of training data. The accuracy and efficiency of the network depends on the network architecture, the optimization of the individual weights and the choice of error function used in the applied optimization procedure.

The design and architecture of the ANN and the subsequent training procedure both follow the approach outlined in [38]. Assume that the vectors $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ contain the neuron variables of the input layer, output layer and hidden layer, respectively. The output layer and hidden layer values can be calculated by the expressions

$$\mathbf{y} = \mathbf{W}_O^\top \mathbf{z}, \qquad \mathbf{z} = \tanh\left(\mathbf{W}_I^\top \mathbf{x}\right), \qquad x_0 \equiv z_0 \equiv 1, \tag{1}$$

where $\mathbf{W}_I$ and $\mathbf{W}_O$ are arrays that contain the neuron connection weights between the input and the hidden layer and the hidden and the output layer, respectively. By having both x_0 and z_0 defined equal to unity biases in the data can be absorbed by the input and hidden layers.

Between the layers a nonlinear transfer function is needed in case of nonlinear output. Tangent hyperbolic has shown to give rise to fast conversion during training [37] and is therefore used throughout this thesis.

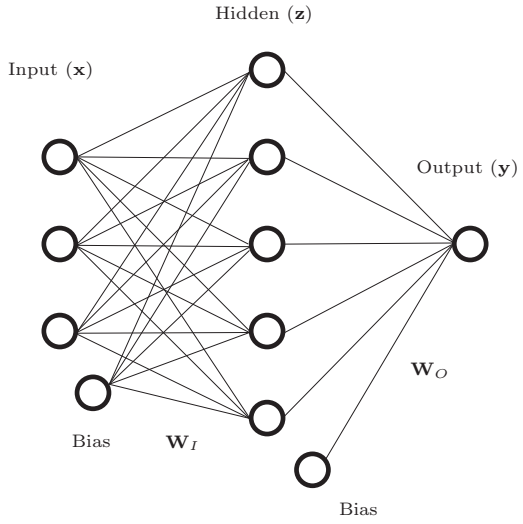


Figure 3: Sketch of artificial neural network.

2.2 Training the ANN

The optimal weight components of the arrays $\mathbf{W}_I$ and $\mathbf{W}_O$ are found by an iterative procedure, where the weights are modified to give a minimum with respect to a certain error function. The error function can be considered as a landscape in a multi dimensional error space. In this error space each connection in the ANN represents a dimension. For regressions models the error function which is minimized during training, is often the mean square error, written as

$$E\left(\mathbf{W}\right) = \frac{1}{2}\sum_{n=1}^{N}\left\{y(\mathbf{x}_n;\mathbf{W}_I;\mathbf{W_O}) - \tau_n\right\}^2 + \frac{1}{2}\alpha\mathbf{W}^2, \tag{2}$$

where y is the ANN output, N is the number of training data sets, τ is the target value for a given input $\mathbf{x}$. The weight decay α in (2) controls the value of the weights and prevents the ANN from overfiting due to noise in the training data. Since the ANN in all cases presented in this thesis simulate theoretical models of physical systems and therefore replicate exact solutions to the systems equations of motion there is no noise in the target output data. Thus, the weight decay in (2) is set to zero ($\alpha = 0$) in the following.

The updating of the weight components is performed by a classic gradient decent technique, which adjusts the weights in the opposite direction of the gradient of the error function [39]. For both $\mathbf{W}_I$ and $\mathbf{W}_O$ of the ANN this gradient decent updating can be written as

$$\mathbf{W}_{new} = \mathbf{W}_{old} + \Delta\mathbf{W}, \qquad \Delta\mathbf{W} = -\eta\frac{\partial E\left(\mathbf{W}\right)}{\partial\mathbf{W}}, \tag{3}$$

where η is the learning step size parameter. The two weight arrays $\mathbf{W}_I$ and $\mathbf{W}_O$ must be evaluated separately yielding these weight updates

$$\Delta W_j^O = -\eta \left(\sum_n^N (y_n - \tau_n)\, z_j \right) \tag{4}$$

$$\Delta W_{ji}^I = -\eta \left(\sum_n^N \left((1 - z_j^2) W_j^O (y_n - \tau_n) \right) x_i \right) \tag{5}$$

The performance of a trained ANN is usually measured in terms of the so-called validation error, which is calculated in the same way as the training error but with respect to an entirely new set of data, which has not been part of the network training.

An ANN with one hidden layer, a sigmoid activation function and a sufficient number of neurons in the hidden layer is an universal approximater because it can approximate any continuous function to any degree of accuracy [40]. However, the purpose of the present ANN is not to replicate a specific set of data but instead to recognize and learn the underlying pattern in the data. That is why it is needed to save some data for validation and to keep track of the validation error during the training. This validation error must be monitored during training of the ANN. The reason is that even though the objective of the training is to minimize the training error, the ultimate goal is to get the validation error as low as possible.

3 Hybrid method

The purpose of the hybrid method is to avoid the task of solving computationally demanding nonlinear differential equations. This is done by training an ANN to simulate a solution to structural equations of motion. This means that the ANN relies on data in order to learn a specific pattern. Hence, the ANN can not stand alone as there must be a source of relevant data. This is why the method described in this thesis is a so-called hybrid method. It is 'hybrid' simply because it combines two different mathematical tools. The training of the ANN depends on data that contains the relation between the external loading on the structure and the corresponding structural response. The data may originate from measurements, experiments or from solutions to numerical models, as long as the data contains the dynamic characteristics of the structure in question. In this case, as in most cases, numerical FEM models are used as data generators.

Since the ANN depends on training data and the numerical model therefore must be established anyway, the hybrid method is mainly relevant in cases where long time series simulations are required. So when evaluating whether or not to use the hybrid method one must assess the balance between the time spend on training the ANN and the time saved once the ANN is ready to produce the simulations. In most cases the time spend on training is insignificant compared to the subsequent required amount of simulation time, while the time

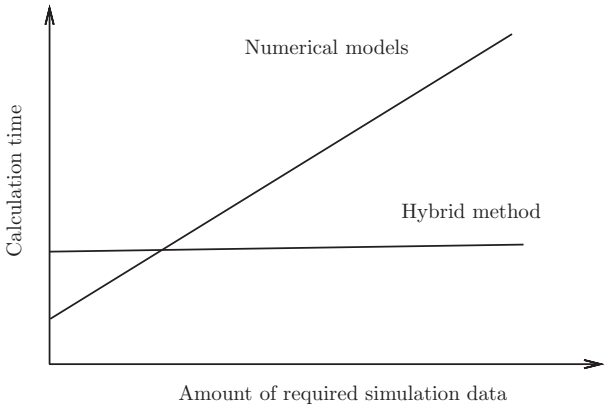


Figure 4: Computational effort associated with nonlinear structural dynamic simulation using the different methods.

saved on the simulations obviously depends on the amount of simulated data, as illustrated in Figure 4.

3.1 Combining with structural models

The first step of the hybrid method is to set up a numerical model of the structure in question, e.g. a FEM model. Having this model it is possible to apply relevant loading and thereby to generate a series of data that subsequently can be used to train the ANN. After the training the ANN is able to take over the simulations at a much higher pace than the corresponding numerical model.

Most of the examples presented throughout this thesis are based on the same numerical model. The model simulates a floating offshore platform anchored by 18 mooring lines assembled in four main clusters. The model calculates tension forces in selected mooring lines as function of prescribed platform motion time histories. A schematic drawing of the floating platform is shown in Figure 5. The external forces acting on the structure are induced by waves, current and wind.

The platform motion and mooring line tension forces are mutually dependent. To simplify the computations the response problem is decoupled. First the platform motions are calculated using a simple quasi-staic catenary mooring line model. Secondly, the dynamic mooring line tension forces are more accurately calculated using a nonlinear finite element mooring line model with prescribed platform motion derived from the precalculated platform motions. The platform motion calculations are carried out by the program SIMO [41] and the dynamic response analysis of the highest stressed line is carried out by the program RIFLEX [42]. The platform mooring line model has been used in appended papers [P1], [P2], [P3] and [P4].

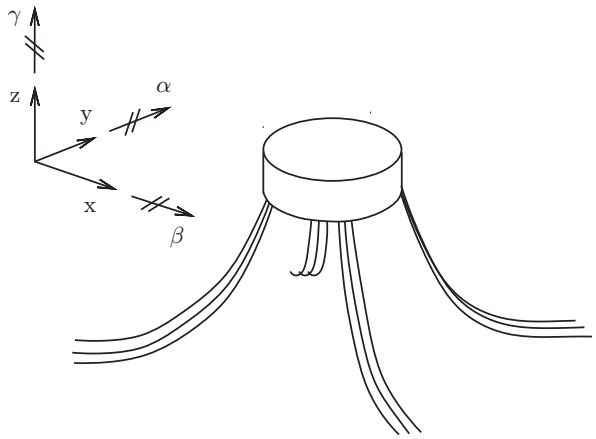


Figure 5: Mooring system with floating platform and anchor lines.

Beside applying the hybrid method to the model of the mooring line on the floating platform the method has been tested on two typical riser structures. In order to do that two simplified FEM models, representing the different typical riser configurations sketched in Figure 6, were set up. The models were established in order to demonstrate that a sequence of neural networks can be used to simulate all critical locations along the complex structure. To conduct the FEM simulations a small tool box, using a co-rotational beam element formulation has been developed. The implementation follows the procedure described in [43] and the riser models have been used in [P5].

4 Mooring line fatigue analysis

The objective of the mooring line analysis in [P1] has been to demonstrate how the training data can be selected and arranged so that a single ANN be trained to simulate an entire scatter diagram, which lists the annual occurrence of all fatigue relevant sea states in terms of significant wave height and wave peak period [44, 45].

The ultimate purpose of the ANN is to completely bypass the computationally expensive numerical time integration procedure, which in this case is conducted by the RIFLEX model. This means that the input to the neural network must be identical to the input used for the RIFLEX calculations. In this case the input is therefore the platform motion, represented by the six degrees of freedom representing the network input in Figure 7 and illustrated by the six vectors in Figure 5. In principle the number of neural network output variables can be chosen freely, and in fact all degrees of freedom from the numerical finite element analysis may in theory be included as output variables in the corresponding ANN. However, the strength of the ANN in this context

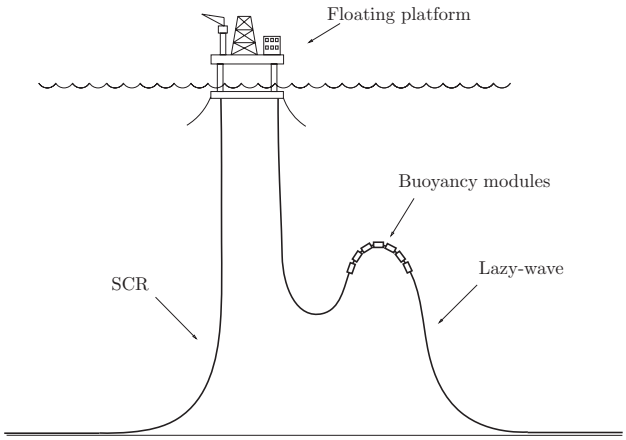


Figure 6: Sketch of two typical riser configurations.

is that it may provide only the specific output variable that drives the design of the structure, which in this case is the maximum top tension forces in the particular mooring line. This leads to a very fast simulation procedure, which for a well-trained network provides sufficiently accurate results. Thus, the ANN is in the present case designed and trained to predict the top tension force of the selected mooring line, and the platform motion (six motion components; surge, sway, heave, roll, pitch and yaw) are, together with the top tension of previous time steps, used as input to the ANN. This is indicated in the network architecture in Figure 7. This means that the input vector $\mathbf{x}_n$ at time increment n can be constructed as

$$\begin{aligned}\mathbf{x}_n = \Big[&[x_t \quad x_{t-h} \dots \ x_{t-dh}] \quad [y_t \quad y_{t-h} \dots \ y_{t-dh}] \quad [z_t \quad z_{t-h} \dots \ z_{t-dh}] \\ &[\alpha_t \quad \alpha_{t-h} \dots \ \alpha_{t-dh}] \quad [\beta_t \quad \beta_{t-h} \dots \ \beta_{t-dh}] \quad [\gamma_t \quad \gamma_{t-h} \dots \ \gamma_{t-dh}] \\ &[T_{t-h} \quad T_{t-2h} \dots \ T_{t-dh}]\Big]^T ,\end{aligned} \tag{6}$$

where $t = nh$ denotes current time, h is the time increment and d is the number of previous time steps included in the input, i.e. the model memory. The corresponding ANN output is the value of the top tension force T_t in the mooring line

$$y_n = T_t \tag{7}$$

Since there is only one network output y is a scalar and not a vector as in (1).

4.1 Selection of training data

Since the ANN is a tool for pattern recognition it can in principal only predict patterns similar to those used for the training of the network. Because of the need for many realizations of time series with different wave conditions in fatigue

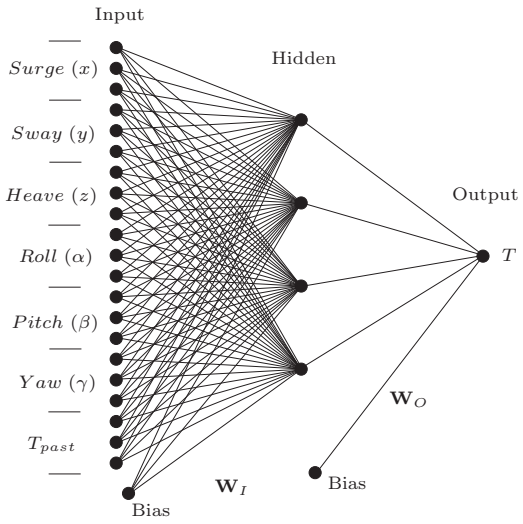


Figure 7: Sketch of artificial neural network for predicting top tension force in mooring line.

analysis the ANN must be trained to cover a broad range of wave characteristics and sea states. This means that the training data must be selected such that in particular all extreme conditions are included but at the same time comprise enough of data in between the extremes to secure a satisfactory representation of the different levels of nonlinear behavior. The sea states used for training in this example are shown in the scatter diagram in Table 1. In the table 'X' denotes fatigue relevant sea states and 'O' represents sea states selected for network training. Note that the sea state with $H_s = 15$m and $T_p = 17.5$s is in fact not part of the fatigue calculations and therefore not part of the original scatter diagram. This specific sea state is merely included to stretch the operational range of the ANN in order to improve the accuracy for the largest relevant wave heights.

The selected training data is assembled in a single sequence of time histories and normalized in the following way,

$$\mathbf{X}_{ANN} = \frac{\mathbf{X}_{FEM} - \mu(\mathbf{X}_{FEM})}{\sigma(\mathbf{X}_{FEM})}, \quad \hat{\mathbf{T}} = \frac{\mathbf{T}_{FEM} - \mu(\mathbf{T}_{FEM})}{\sigma(\mathbf{T}_{FEM})} \tag{8}$$

The normalization of data leads to faster training and better convergence as described in most text books on neural networks, see e.g. [46]. After the normalization the data has a mean value of $\mu(\mathbf{X}_{ANN}) = 0$ and a standard deviation of $\sigma(\mathbf{X}_{ANN}) = 1$. A similar normalization is applied for the network input that goes into the ANN after training in the subsequent simulations.

Note that in order to obtain the actual tension force, $\hat{\mathbf{T}}$ must be transformed back using $\mu(\mathbf{T}_{FEM})$ and $\sigma(\mathbf{T}_{FEM})$ calculated during normalization of

H_s/T_p	1.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5	10.5	11.5	12.5	13.5	14.5	15.5	16.5	17.5	18.5	19.5	20.5	21.5	22.5
1	X	X	X	X	X	X	X	X	X	X	X	X	O	X	X	X	X	X	O		
2	O	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
4			X	X	X	X	X	X	O	X	X	X	X	X	X	X	X	X			
5				O	X	X	X	X	X	X	X	X	X	X	X	X	X				
6					X	X	X	X	X	X	X	X	X	X	X	O					
7						X	X	X	X	X	X	O	X	X	X						
8							O	X	X	X	X	X	X	X	X						
9								X	X	X	X	X	X	X	X						
10									X	X	X	X	X	X	X						
11										X	X	X	X	O	X						
12											O	X	O	X	O						
13												X	O	O							
15																O					

Table 1: Scatter diagram of relevant sea states. Training data are marked by 'O'.

the training data in (8).

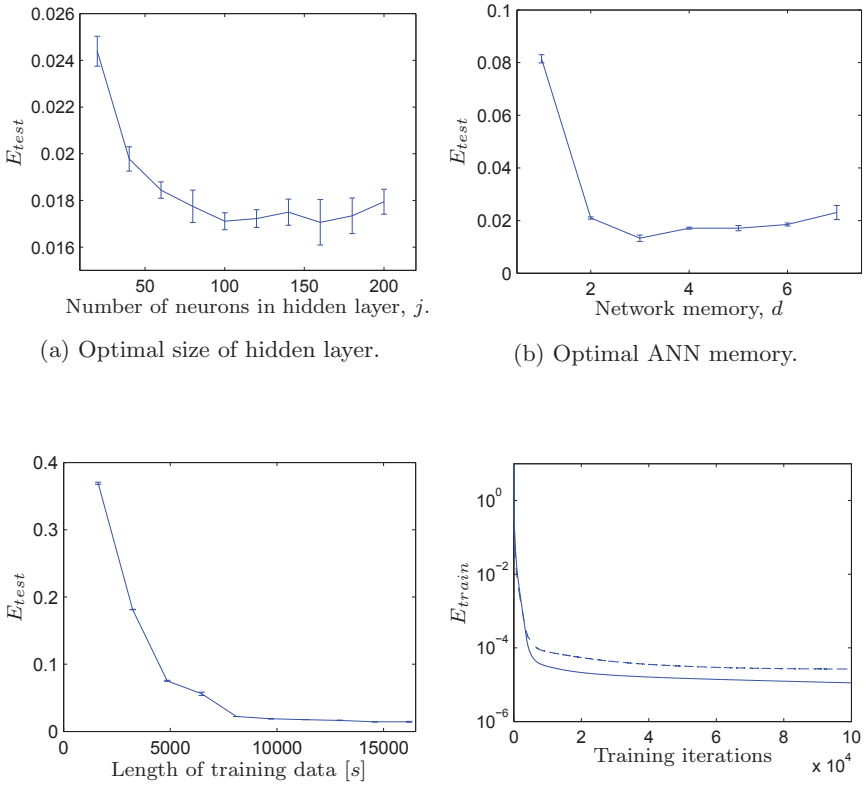
4.2 ANN design

In order to obtain an optimal ANN structure three variables have been investigated. The ANN has been optimized with respect to the number of neurons in the hidden layer, the model memory and amount of training data. The model memory represents the number of previous time steps used as ANN input. As mentioned previously all data generated by the RIFLEX model must be divided into two sets - a training set and a validation set. The validation set is used to asses the quality of the trained ANN with respect to fresh data. The validation error E_{test} is calculated in the same way as the training error (2) but with respect to data that has not been part of the initial ANN training.

Figures 8a-8c show the results of the three investigations. In all figures the validation error E_{test} is plotted as function of the investigated free parameter. All curves represent the mean value of the error from five independent simulations, and the vertical bars show the corresponding standard deviation. Based on these results an ANN with 100 neurons in the hidden layer and a model memory of three time steps has been chosen in the subsequent simulations. The training is in this case based on $1.6 \cdot 10^4$ s of simulated data divided equally on the 15 selected sea states.

4.3 Assessment of simulation accuracy

One of the challenges associated with the hybrid method is the assessment of when the ANN simulations are sufficiently accurate. The error function obviously provides a measure for the ANN accuracy but it is difficult to evaluate how well the ANN actually performs based on this parameter. Figure 8d shows the development in ANN error during training. The training error E_{train}, which is minimized by the training algorithm, is represented by the solid curve, while the dashed curve shows the ANN validation error E_{test}. This latter measure indicates how well the ANN performs with respect to fresh data. Since the goal of the procedure is to train the ANN to actually simulate the structural response with respect to unknown data, the underlying objective of the training



(a) Optimal size of hidden layer.

(b) Optimal ANN memory.

(c) Optimal amount of training data.

(d) ANN training.

Figure 8: Optimization ANN architecture

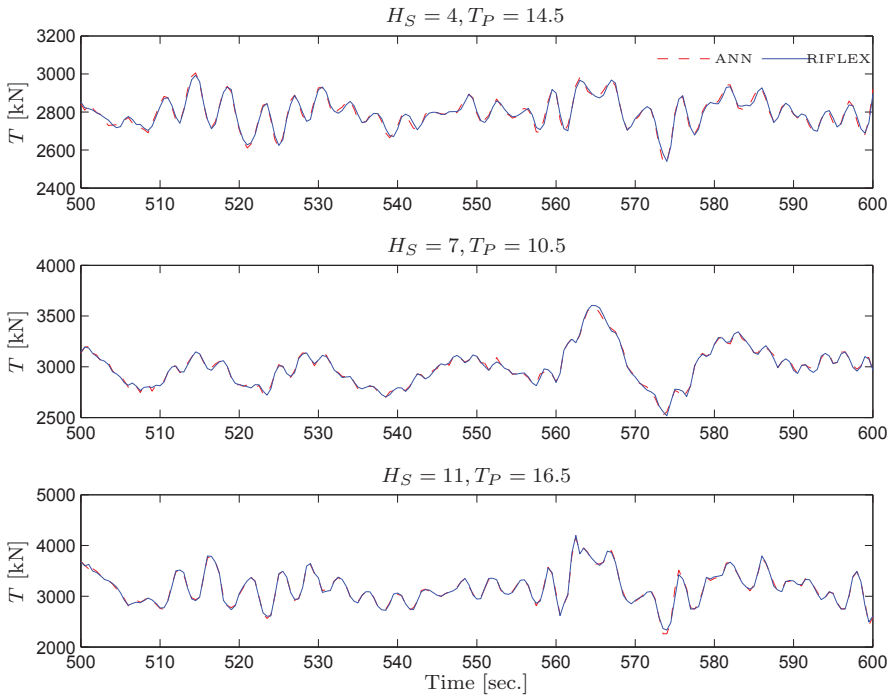


Figure 9: ANN and RIFLEX simulation for time series for 3 different sea states.

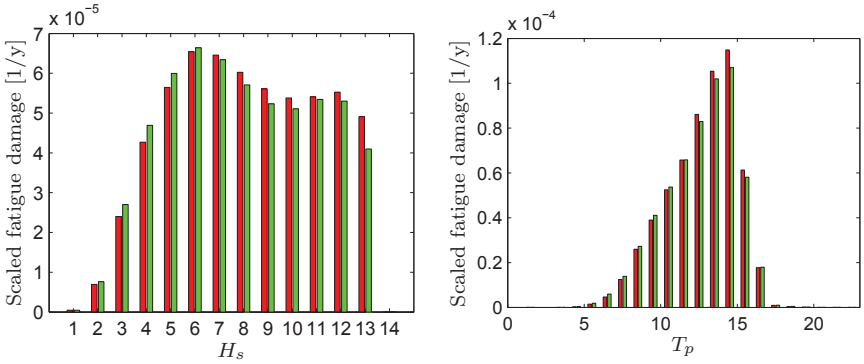
is therefore to minimize E_{test}. Hence, one must keep track of the development of the validation error during the training to avoid overfitting with respect to the specific training data. An increase in E_{test} often indicates overfitting, in which case the training should be terminated. Even though the training reduces the error measures as intended it is difficult to evaluate how accurate the ANN performs based solely on these error magnitudes. Under all circumstance some form of visual inspection will be an important part of the assessment of the ANN performance. Figure 9 shows comparisons of the ANN and the FEM simulations. The figure shows three different sea states that have not been part of the ANN training.

Since these long time domain simulations are often used in fatigue life calculations an appropriate way to asses the accuracy of the ANN is simply to compare the accumulated damage. Therefore, a full life time assessment for the mooring line has been carried out based on all sea states in Table 1. The expected life time is estimated based on the results obtained by the hybrid method simulations and then compared to the estimate obtained by the full RIFLEX analysis. The analysis has been conducted in accordance with offshore standard and recommended practice for riser analysis by DNV [1, 2].

The damage has been calculated by a Palmgren-Miner summation of the

	Fatigue life [years]	Deviation [%]	Simulation time
RIFLEX	1698	-	$\sim$ 10 hours
Hybrid	1725	$\sim$ 1.6 %	$\sim$ 1 minute

Table 2: Calculated expected fatigue life for the anchor line.



(a) Accumulated yearly damage H_S.

(b) Accumulated yearly damage T_P.

Figure 10: Accumulated yearly damage divided into wave heights and peak periods. ANN, RIFLEX

stress variation cycles obtained from a rain flow counting procedure for each simulation time history [47]. The damage from each sea state has been multiplied with the likelihood of the given sea state. This provides the expected annual damage and hence the inverse of the expected fatigue life of the mooring line once the individual contributions are summed over all sea states. The results are listed in Table 2. It is seen that the hybrid method deviates by only 1.6 % from the RIFLEX analysis, which in this particular case must be considered as the correct estimate.

Considering that the hybrid method is about 600 times faster than the RIFLEX analysis the 1.6 % deviation is remarkable. If the length of the time domain simulations is increased the reliability of the analysis is also increased. With the RIFLEX model increasing simulation lengths are quite expensive, whereas with the hybrid method the simulation length does not make any practical difference as the simulations after training are very fast.

Figures 10a and 10b show the accumulated damage for all significant wave heights and wave peak periods, respectively. It is seen that the results of the ANN simulations are very close to those of the RIFLEX simulations on all individual levels.

5 Sequential ANN simulation

The mooring line example presented in section 4 presumably featured a very direct load-response relation between platform motion and the resulting top tension forces in the mooring line. For risers where the design is based on section forces at numerous hot spots along the structure it is not straight forward to predict section forces at the bottom of the riser merely based on the motion at the sea surface. A sequential procedure for the analysis of flexible risers has been proposed in [P5]. The procedure uses a series of ANNs to simulate the response of the entire riser for a given load history. Each ANN simulates the response of an adjacent node. This response is then used as input for a subsequent ANN simulating the response of the next model node. Thereby it is possible to simulate response histories at all critical locations of the flexible risers. A simplified two-dimensional FEM model representing typical riser configurations has been used to demonstrate the sequential ANN scheme. The model does not consider hydrodynamic effects or interaction with the sea bed. Furthermore, the only external impact on the structure is a prescribed horizontal movement of the top node on the riser - corresponding to the movement of a floating platform.

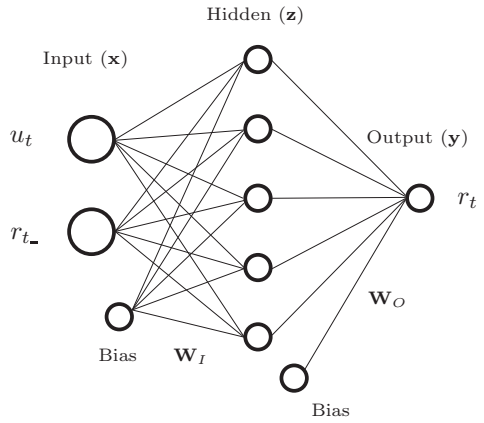


Figure 11: Sketch of artificial neural network.

The ANN used in this example is sketched in Figure 11 and has the exact same structure as the one used for the mooring line simulations in section 4. The ANN is trained to predict the horizontal response of the riser based on the prescribed horizontal movement of the top node. In connection with flexible riser analysis the variables u_t and r_t represent the prescribed floating platform motion and the corresponding structural riser response at time t, respectively. Structural response at previous time steps are denoted $r_{t\_}$.

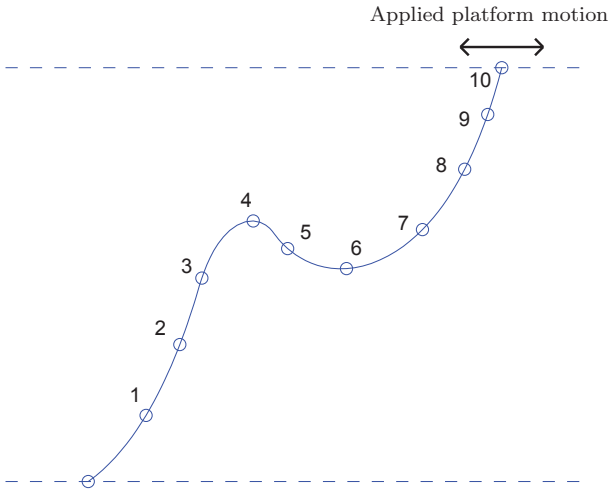
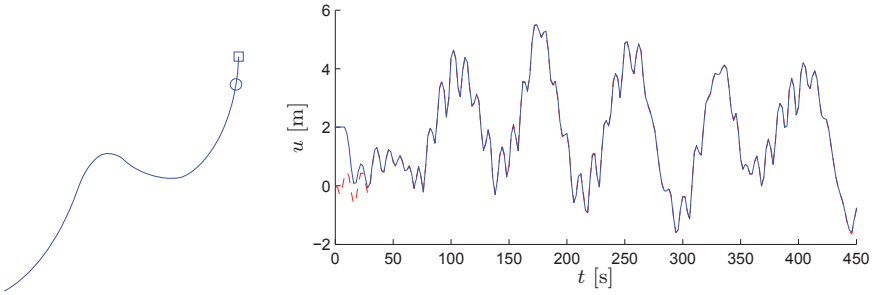


Figure 12: Sketch of FEM model.

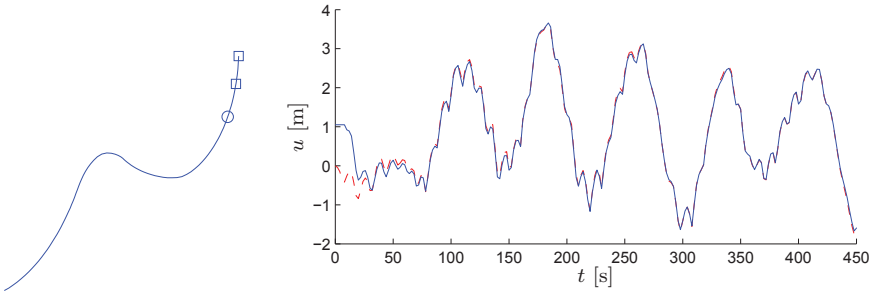
5.1 Applied to riser in lazy wave configuration

The sequential ANN scheme has been demonstrated on two typical riser configurations. In both cases ten model nodes are selected as indicated in Figure 12. The ambition has been to simulate the response of the bottom node (node 1) based only on the prescribed motion of the top node (node 10). The first case is a simple steel catenary riser. In that example it is possible to obtain accurate simulation of the bottom node by use of a series of just two ANNs. The second example, which is also the one included in this thesis, considered the somewhat more complex lazy-wave riser configuration. For this riser configuration it turns out that, in order to keep the simulation error sufficiently low, it is necessary to have a distinct ANN for each of the ten selected nodes in Figure 12. Figure 13 shows how the individual networks perform at different locations along the riser. Except for the first ANN in Figure 13a all networks use simulated input generated by the previous ANN. This means that any previous inaccuracies are passed on and thereby remain in subsequent simulations. Despite the accumulation of errors, the sequential hybrid method still manages to maintain a very high degree of accuracy in the response simulations all the way along the riser down to the final node 1 at the sea bed. Note that each ANN introduces a short transient response period before the response gets synchronized with the input. These transient periods are accumulated trough the sequence which is why it takes about 200 s before the simulation of node 1 in Figure 13c replicates the FEM simulation.

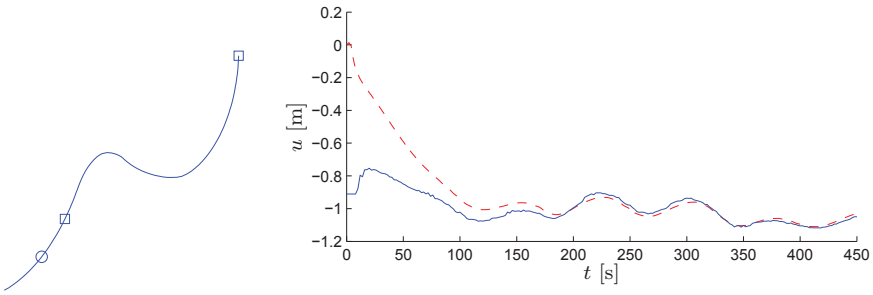
The use of more ANNs obviously implies extra work in establishing the simulation tool. The results of the FEM simulation, which the ANN training is based on, contain the response histories of all model nodes. This means that



(a) Response at Node 9 using Node 10 as input.



(b) Response at Node 8 using Node 9 and 10 as input.



(c) Response at Node 1 using Node 2 and 10 as input.

Figure 13: Comparison of simulations generated by FEM and ANN. □: ANN input, ○: ANN output, ——— ANN, – – – FEM

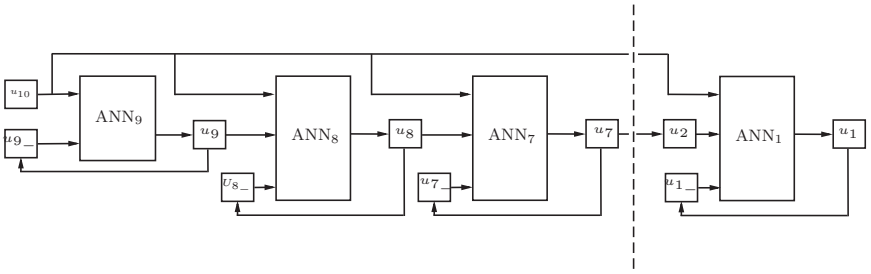


Figure 14: ANN structure for Lazy-wave simulation.

the training of all the networks involved in the simulation can be done based on the same FEM simulation. Hence, the sequential scheme requires extra time for ANN training each time an additional network is introduced, but no extra time spend on the computational demanding nonlinear FEM simulation.

The series of networks which made it possible to simulate the riser response all the way down to the sea bed, stepping through the structure continuously using simulated data one model node as input for a subsequent ANN which simulates the next node is shown in Figure 14. The simulation of the lazy-wave riser includes a total of nine ANNs which must be trained individually. However, considered that all training can be carried out based on just one FEM simulation and that the time spend on ANN simulation after training is insignificant, when compared to the time consuming FEM analysis, the extra training is not considered to be a significant degradation of the hybrid method.

6 Optimization of ANN performance

The following optimizations have been done on the ANN trained to simulate top tension forces in the mooring line as in section 4. For clarity the optimization algorithms were applied to a network which is trained to cover only a few sea states. This reduces the optimal number of neurons in the hidden layer from about one hundred, as necessary when the ANN must cover the entire scatter diagram in Table 1, down to just four, which gives exactly the ANN sketched in Figure 7. In the previous examples the individual neural networks have been optimized with respect to size of hidden layer, model memory and amount of training data, but not with respect to internal structure and the objective of the ANN. This section deals with the possibility of enhancing ANN performance through general optimization approaches.

6.1 Comparison of error functions

The mean square error (MSE) used as error function so far is very convenient and usually leads to good ANN performance. But it does not provide any

Table 3: Deviations on accumulated tension force cycles.

H_S	E_2	E_2^w	E_4	E_4^w
2	−3.6 %	−16.3 %	7.3 %	65.8 %
4	−6.9 %	−13.0 %	4.8 %	32.0 %
6	−4.4 %	−8.0 %	2.6 %	14.9 %
8	−2.7 %	−4.8 %	−1.5 %	8.6 %
10	−2.0 %	−3.5 %	−1.6 %	6.9 %
12	−1.4 %	−3.1 %	−1.7 %	5.3 %
14	−0.8 %	−0.3 %	2.1 %	3.1 %
Total	−1.7 %	−3.8 %	1.6 %	6.4 %

extra focus on specific data. For most cases this is probably an advantage. However, for mechanical structures that is not always the case. The comparison of error measures presented in [P2] yielded two useful contributions. The paper describes an attempt to improve the ANN performance through the use of tailor-made error functions that aim to focus on large response amplitudes as these contribute excessively to the structural damage. The study compared different error measures with respect to accuracy in fatigue calculations. The comparisons were based on the so-called *Minkowski-R* error:

$$E_R = \frac{1}{R} \sum_n \sum_{k=1}^{c} |y_k(\mathbf{x}_n; \mathbf{W}) - t_{kn}|^R \tag{9}$$

where y is the scalar ANN output and t is the target value. The classic MSE is seen to be a special case of the Minkowski error with $R = 2$. Different values of R have been tested together with the possibility of emphasizing the focus on large response amplitudes by putting a weight on the error function as

$$E_R^w = \frac{1}{R} \sum_n \sum_{k=1}^{c} |y_k(\mathbf{x}_n; \mathbf{W}) - t_{kn}|^R \cdot |t_{kn}| \tag{10}$$

The results listed in Table 3 show that raising the power from $R = 2$ to $R = 4$ improves the accuracy slightly while the weighted error function in fact impairs the ANN performance significantly.

In addition to a slight improvement in fatigue life estimation the study provided a beneficial side effect. Raising the power of the error function from the usual two to four provides a function to which the second derivative with respect to the network weight arrays $\mathbf{W}$ is easily calculated. The double differentiation of the error function is important part of the most used ANN optimization algorithms.

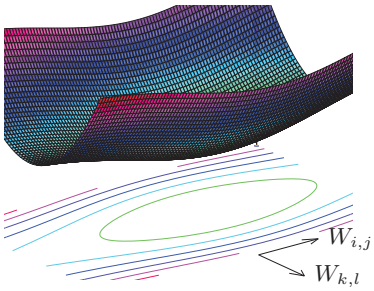


Figure 15: The error function around a local minimum. The curvature of the error surface in each direction corresponds to the saliency of the associated weight. In this illustration we see that $\frac{1}{2}\frac{\partial^2 E}{\partial W_{i,j}^2}W_{i,j}^2 < \frac{1}{2}\frac{\partial^2 E}{\partial W_{k,l}^2}W_{k,l}^2$.

6.2 Optimization of ANN structure

Appended papers [P3,P4] investigate the possibility of improving the ANN performance by optimizing the structure of the network used in [P2]. The aim of the optimization procedures is initially to classify the layer connections in terms of relevance, and subsequently to use this classification to remove connections and thereby improve the efficiency of the ANN. Thus, the optimization procedures initiate from a reasonably large network that has been trained to minimum error. Although the ANN used in the present case already is rather compact in terms of size of input and hidden layer, it is assumed that the network is still sufficiently large to be pruned further on individual weight level.

There exist several methodologies for evaluating the saliency of the network weights [37]. The idea of the pruning methods is to rank the importance of all network weights and then successively delete the least salient weight. The simplest possible approach is to simply consider the magnitude of all network weights and assume that small weights are less important than large weights. However, according to [37] this concept has little theoretical motivation. Nevertheless, for comparison this method has been implemented and tested in the following together with the two most renown optimization algorithms Optimal Brain Damage (OBD) [27] and Optimal Brain Surgeon (OBS) [28].

The starting point for both OBD and OBS is a Taylor expansion of the error function (9) with respect to the network weights. This Taylor series can be written as

$$\delta E = \frac{\partial E}{\partial \mathbf{W}^T}\,\delta\mathbf{W} + \frac{1}{2}\delta\mathbf{W}^T\mathbf{H}\,\delta\mathbf{W} + O(\|\delta\mathbf{W}\|^3), \tag{11}$$

where

$$\mathbf{H} = \frac{\partial^2 E}{\partial\mathbf{W}\partial\mathbf{W}^T} \tag{12}$$

is the Hessian matrix containing all second order derivatives of the error function with respect to the network weights. Both optimization methods are based on

networks that are initially trained to a local minimum. This means that the first term in (11) vanishes because $\partial E/\partial \mathbf{W}^T = \mathbf{0}$ represents the condition for a function extremum. The higher order terms represented by the latter term in (11) are furthermore omitted. This implies that the saliency of the network weights is calculated merely by evaluating the second order derivatives, whereby the problem reduces to

$$\delta E \simeq \frac{1}{2}\delta \mathbf{W}^T \mathbf{H}\, \delta \mathbf{W} \tag{13}$$

This approximation of the perturbation of the error function gives the curvature in error space at the local minimum that was reached during the network training. The particular diagonal element in the Hessian matrix with smallest magnitude identifies the weight component with least significance with respect to the error function. This weight is therefore deleted by the pruning procedure. Figure 15 shows a two-dimensional example where the weight $W_{i,j}$ is less significant compared to $W_{k,l}$ because $\frac{1}{2}\frac{\partial^2 E}{\partial W_{i,j}^2} W_{i,j}^2 < \frac{1}{2}\frac{\partial^2 E}{\partial W_{k,l}^2} W_{k,l}^2$.The expression in (13) constitutes the foundation for both the OBD and the OBS procedure, even though the two methods use different approaches to determine the Hessian matrix $\mathbf{H}$, as demonstrated next.

6.2.1 Optimal Brain Damage

The OBD procedure trains the initial ANN architecture, where all neurons between input and hidden layer and between hidden and output layer are connected with non-vanishing weights. The saliency of all weights are computed and the least important weight is pruned by simply defining the associated weight to be zero. When a network weight is eliminated the reduced network must be retrained before the next least salient weight can be selected and subsequently removed. The procedure can in principal be repeated until the network is completely eliminated, whereas in reality the procedure is terminated when a desired compromise between accuracy and computational efficiency is obtained.

The OBD procedure approximates the Hessian by disregarding the non-diagonal terms. With the error function given in (9) and $R = 4$ this assumption yields the following second derivatives of the network error with respect to the components of the weight matrices $\mathbf{W}_O$ and $\mathbf{W}_I$, respectively. The diagonal Hessian components for the hidden-to-output layer connections can be computed as

$$\frac{\partial^2 E}{\partial W_{O,j}^2} = 3\sum_{n=1}^{N} (y_n - t_n)^2 z_j^2 \tag{14}$$

while for the input-to-hidden layer connections

$$\frac{\partial^2 E}{\partial W_{I,ji}^2} = \sum_{n=1}^{N} (1 - z_j^2) \left(3(1 - z_j^2)(y_n - t_n)^2 W_{0,ji} - 2z_j(y_n - t_n)^3\right) W_{0,ji} x_i^2 \tag{15}$$

Having these double derivatives determined by (14) and (15) it is possible to evaluate the saliency for all individual network weights as

$$L_q^{OBD} = \frac{1}{2}\mathbf{H}_{qq}W_q^2, \tag{16}$$

where the subscript represents the q'th weight. Thus, the least salient weight can be identified by L_q^{OBD} in (16) and subsequently deleted.

6.2.2 Optimal Brain Surgeon

The goal of the OBS method is to minimize the increase in the error function (9) while setting a specific network weight to zero. The vanishing weight, which represents the candidate for deletion, is in the following denoted W_q. The condition for elimination of the weight with the smallest saliency can be expressed by the projection

$$\mathbf{e}_q^T\delta\mathbf{W} + W_q = 0 \tag{17}$$

where $\mathbf{e}_q$ is a zero vector in weight space with unit value corresponding to weight W_q. The aim is now to solve the following equation, which minimizes the increase in error while at the same time forcing the selected weight to zero,

$$\min_q \left\{ \min_{\delta\mathbf{W}} \left(\tfrac{1}{2}\delta\mathbf{W}^T\mathbf{H}\,\delta\mathbf{W} \right) \mid \mathbf{e}_q^T\delta\mathbf{W} + W_q = 0 \right\} \tag{18}$$

To solve (18) a Lagrangian is formed containing $\delta E = 0$, with the perturbation of the error function given in (13), and the condition in (17) enforced by the Lagrange multiplier λ,

$$L = \frac{1}{2}\delta\mathbf{W}^T\mathbf{H}\,\delta\mathbf{W} + \lambda(\mathbf{e}_q^T\delta\mathbf{W} + W_q) \tag{19}$$

When deriving the functional derivatives of this Lagrangian, and employing the constraint condition in (17), the perturbation of the weight matrix is obtained as

$$\delta\mathbf{W} = -2\lambda\mathbf{H}^{-1}\mathbf{e}_q \tag{20}$$

This is now inserted into (17), which then gives the Lagrange multiplier λ as

$$\lambda = \frac{W_q}{2\,[\mathbf{H}^{-1}]_{qq}} \tag{21}$$

where $[\mathbf{H}^{-1}]_{qq}$ represents the diagonal element of the inverse Hessian corresponding to the weight component q. To calculate the optimal change in weight the value of λ is finally inserted into (20), which gives

$$\delta\mathbf{W} = -\frac{W_q}{[\mathbf{H}^{-1}]_{qq}}\,\mathbf{H}^{-1}\mathbf{e}_q \tag{22}$$

Now that this optimal change in weight has been determined the resulting change in the error is obtained by inserting (21) and (22) into (19),

$$L_q^{OBS} = \frac{1}{2}\frac{W_q^2}{[\mathbf{H}^{-1}]_{qq}} \tag{23}$$

As seen the inverse Hessian is needed for calculating both the smallest saliency L_q^{OBS} and the corresponding optimal change in weight $\delta\mathbf{W}$. Hassibi and Stork [28] outlines a procedure for the direct evaluation of the inverse Hessian matrix $\mathbf{H}^{-1}$, which avoids the computational effort associated with inversion of large matrices. However, since the weight matrices used in the example for the present paper are fairly small, the inversion of $\mathbf{H}$ is based on standard numerical techniques without special considerations.

6.3 Application to structural model

The three optimization algorithms have been applied to the trained ANN. Figure 16 shows the development of the ANN error as the number of deleted network weights increases. To verify that the three procedures actually select and delete appropriate weights a fourth algorithm, which randomly deletes weights, is also included in Figure 16. For the simple approach, OBD and random deletion of weights the network is retrained each time a weight is deleted whereas for the OBS the network is trained only once, whereafter the network weights are adjusted according to the procedure described in Section 6.2.2 and therefore this procedure requires no additional network training. Apparently this procedure for updating weights without additional training holds only for a limited number of deleted weights in this particular example.

The initial network training requires a fairly large amount of iterations $(1 \cdot 10^5)$ to reach a local minimum in error. However, the retraining of the network performed after each weight deletion does not have to be as thorough. In this example 2000 iterations for the retraining have proven to give satisfactory results, which means that the retraining is not very time consuming. Hence, the retraining may therefore be considered as a small adjustment to the weights and not a complete network training. Even though retraining should be unnecessary in the OBS procedure it is recommended that the final network is retrained after the pruning exercise has been terminated. The performance of the networks pruned by the OBS procedure is shown in Figure 16 by the green dashed curve. It is seen that the error E of the OBS procedure is relatively low for up to approximately 25 pruned weights. Although simulations presented by Hassibi and Stork [28] indicate that the OBS procedure is superior to OBD, it appears from the present analysis that the OBD procedure is apparently the best pruning procedure in this particular example concerning dynamic analysis of a slender off-shore structure. On the other hand, the very simple approach, where weights are directly deleted according to their magnitude, seems to work very well in this particular example despite its alleged poor performance in practice [37].

Figure 17 shows just how well the OBD procedure works for this example maintaining a very high simulation accuracy after having pruned the vast majority of the networks weights.

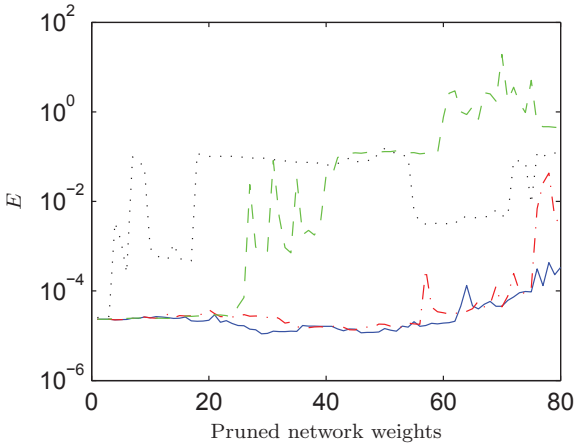


Figure 16: Development in error E as number of pruned network weights increases. Simple, OBD, OBS, Random.

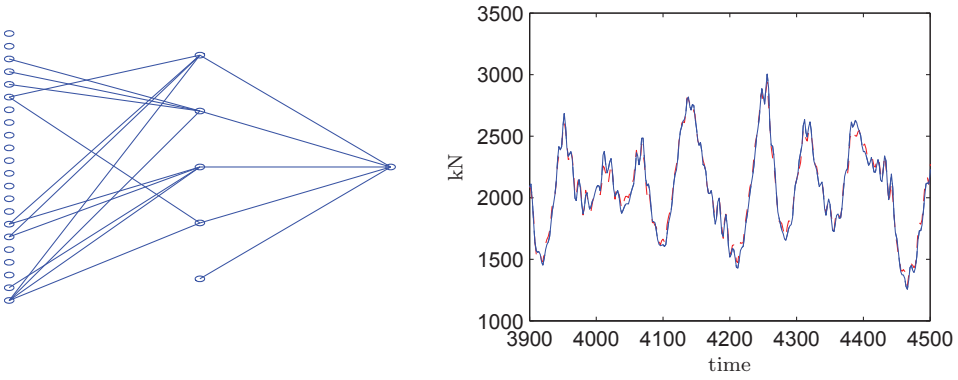


Figure 17: Network with three input components pruned (heave, roll and pitch) - 74 weights deleted by OBD. ANN, RIFLEX

6.4 Redundant input components

As the pruning procedure successively cuts network weights until the entire network is deleted it will at certain stages reach a point where a motion degree of freedom for the platform is completely ignored by the network. This can be used to rank the importance of the individual motion components and to evaluate whether one or more components are in fact negligible. This feature is addressed in [P4]. When the trained ANN is pruned by the OBD procedure the optimal network emerges around the same stage as the heave motion of the platform is completely removed from the network input. The heave motion component is completely ruled out after 37 deletions. The fact that the optimal ANN is reached as the heave motion is ruled out indicates that not only is the heave motion redundant it actually adds noise to the system. The explanation for this is that the numerical model has a build-in heave compensator that dampens the effect of the vertical platform motion. Hence, the tension forces in the mooring line is not directly related to the heave motion of the floating platform. The next degree of freedom to be completely ignored by the network is the platform roll motion. This happens after 53 deletions. After 74 deletions the third motion degree of freedom is ignored by the network. At this stage it is the platform pitch motion which no longer plays a part in the ANN simulation. Visual inspection of the simulations in Figure 17 shows that the ANN still simulates quite accurately at this stage. However, the graphs included in [P4] reveal that continuing the pruning until a fourth motion component (surge) is deleted is going to far. At this stage the network still captures the overall dynamics of the structure, but it is no longer able to do any useful simulation.

As for the the example in section 4 the performance of the individual ANNs are compared with respect to accuracy on summarized stress cycles using rain flow counting. The summation is done for all seven sea state simulations. Deviations from the RIFLEX simulation are listed in Table 4. The table lists results at the different stages of pruning where a motion degree of freedom is ignored by the ANN. To test how well the OBD works as a tool for quick parameter study, a series of RIFLEX simulations are conducted in accordance with the ranking given by the OBD procedure. Hence, the platform heave motion is left out of the first calculation, heave and roll left out in the second calculation and so forth. The deviations in accumulated stress cycles are listed in Table 5. It is seen that leaving out the heave motion does not degrade the accuracy of the simulation significantly. When leaving out heave and roll motion from the analysis, the error increases slightly - as seen for the ANN simulation. For some reason, when the pitch motion is ignored the error in the stress cycle accumulation drops to practically nothing. This is in contrast to the ANN simulations where a general increase in error is seen as the number of input variables are ignored. A closer look, however, at the data (not included here) reveals that the surprisingly low deviation in accumulated load stress cycles is coincidental. When all four input variables are ignored (heave, pitch, roll and surge) the simulation is completely useless. The inaccuracy of accumulated load stress cycles increases dramatically. Except for the very low error in the fatigue estimation based on the simulation

Table 4: Deviations on accumulated tension force cycles when compared to RIFLEX simulations.

Simulation	ANN	ANN_1	ANN_2	ANN_3	ANN_4
Deviations	1.2 %	−1.1 %	−3.4 %	−8.8 %	7.7 %

Table 5: Deviations on accumulated tension force cycles when compared to RIFLEX simulations.

Simulation	RIFLEX	$RIFLEX_1$	$RIFLEX_2$	$RIFLEX_3$	$RIFLEX_4$
Deviations	-	-0.5 %	3.1 %	0.03 %	-65.4 %

conducted with RIFLEX ignoring heave, pitch and roll, the results obtained by the two methods coincides very well. This implies that, based on just one relatively short simulation sequence including three different sea states, the trained ANN combined with the OBD procedure is able to evaluate the importance of the input variables and to estimate the cost of ignoring one or more variables in the analysis including seven different sea states. In addition, the pruning procedure detects the stage where the simulation becomes completely useless.

7 Conclusion

The goal for this PhD study has been to develop the hybrid scheme to a point were it is ready for commercial applications used in design and analysis of slender marine structures. Even though this goal has not been reached completely the results obtained and presented throughout this thesis indicates that hybrid method simulation holds a great potential. Three main contributions have moved the method closer to a application ready stage.

First there is the novel way of selecting and arranging training data which makes a single ANN able to simulate all relevant sea states for an entire fatigue analysis. The example presented in [P1] showed that the time spend on time domain simulations conducted as part of a fatigue analysis on the mooring lines on a floating offshore platform could be reduced from about 10 hours to less than 2 minutes while maintaining an accuracy on the fatigue life estimation of about 98%.

The second extension of the hybrid method is the sequential ANN procedure that makes it possible to simulate all critical locations and hot spots on complex structures. This procedure introduced a series of mutually dependent ANNs and thereby additional time spend on network training. However, since this

extra training do not require any additional time consuming FEM analysis, extra training does not significantly degrade the potential of sequential hybrid method simulation. Of the contributions presented in this thesis, this sequential method is probably the one which needs most further development before being ready for implementation in commercial software.

The third development is the application of optimization algorithms for ranking and evaluating the importance of external effects on structures. It was shown that the OBD procedure effectively ranked the importance of the platform motions and also with great accuracy identified insignificant components. Noting that training and pruning an ANN is a lot less time consuming than running several numerical simulations, the described procedure can become a very valuable tool when investigating the importance of various input variables in mechanical analysis.

The application of artificial neural networks for dynamic simulation of slender marine structures is not fully develop to the extent where it can be implemented directly into commercial software. However, it is probably closer to being ready than the most potential users are. Machine learning techniques are not part of the education for mechanical engineers - and it most probably won't be in any near future. In addition, when compared to regular mathematical tools applied in mechanical engineering, artificial neural networks are still 'black boxes' when it comes to interpretation of output data. Therefore, one of the challenges with the hybrid method is to implement it in a way so that it can be used reliably by engineers who do not possess in-depth knowledge about these techniques. Hence, the programs must be set up in a way so the architecture, optimization and validation to a large extent is automated. However, if this is done properly the method potentially can save a tremendous amount of calculation time in nonlinear dynamic analysis.

References

[1] DNV. *Offshore Standard DNV-OS-F201 - Dynamic Risers*. Det Norske Veritas, October 2010.

[2] DNV. *Recommended Practice DNV-RP-F204 - Riser Fatigue*. Det Norske Veritas, July 2005.

[3] API. *API RP 16Q - Recommended Practice for Design, Selection, Operation and Maintenance of Marine Drilling Riser Systems*. American Petrolium Institue, August 2010.

[4] N. Sødahl and C.M. Larsen. Methods for estimation of extreme repsonse of flexible risers. *Proceedings of the Second International Offshore and Polar Engineering Conference*.

[5] L.V.S. Sagrilo, Z. Gao, A. Naess, and E.C.P. Lima. A streightforward approach for using single time domain simulations to assess characteristic response. *Ocean Engineering*, 38(5):1464–1471, August 2011.

[6] A.O. Vazquez-Hernandez, B.B. Ellwanger, and L.V.S. Sagrilo. Longterm response analysis of fpso mooring systems. *Applied Ocean Research*, 33(4): 375–383, October 2011.

[7] L.V.S. Sagrilo, A. Naess, and A.S. Doria. On the long-term response of marine structures. *Applied Ocean Research*, 33(3):208–214, July 2011.

[8] S.R. Winterstein, T.C. Ude, C.A. Cornell, P. Bjerager, and S. Haver. Environmental parameters for extreme response: Inverse form with omission factors. *ICOSSAR-93*.

[9] H. Adeli. Neural networks in civil engineering: 1989-2000. *Computer-Aided Civil and Infrastructure Engineering*, 16:126–142, 2001.

[10] Z. Waszczyszyn and L. Ziemiański. Neural networks in mechanics of structures and materials - new results and prospects of applications. *Computers and Structures*, (79):2261–2276, January.

[11] S. Gupta, A. Ray, and E. Keller. Symbolic time series analysis of ultrasonic data for early detection of fatigue damage. *Mechanical Systems and Signal Processing*, 21:866–884, 2005.

[12] Javier Valentin-Sivico, Vittal S. Rao, and Vasudha Samanthula. Damage detection of bridge-like structures using neural networks. *Smart Structures and Materials: Smart Systems for Bridges, Structures, and Highways - Proceedings of SPIE*, 3325:139–148, 1998.

[13] X. Fisher K. Ordaz-Hernandez and F. Bennis. Modal reduction technique for mechanical behaviour modelling: Efficiency criteria and validity domain assessment. *Journal of Mechanical Engineering Science*, 222:493–505, 2007.

[14] B. Xu, Z. Wu, and K. Yokoyama. Neural network based modeling of structural parametric evaluation with dynamic respones in time domain. *Smart Nondestructive Evaluation and Health Monitoring of Structural and Biological Systems II - Proceedings of SPIE*, 5047:392–402, 2003.

[15] R. Guarize, N.A.F. Matos, L.V.S. Sagrilo, and E.C.P. Lima. Neural networks in dynamic response analysis of slender marine structures. *Applied Ocean Research*, (29):191–198, January 2008.

[16] R.W. Clough and J. Penzien. *Dynamics of Structures*. McGraw-Hill, 1993.

[17] J.W. Tedesco, W.G. McDougal, and C.A. Ross. *Structural Dynamics - Theory and Applications*. Addison-Wesley, 1999.

[18] K. Gurney. *An Introduction to Neural Networks*. CRC Press, 1997.

[19] S. Rajasekaran and G.A. Vijayalakshmi Pai. *Neural Networks, Fussy Logic, and Genetic Algorithms*. PHI Learning Private Limited, 2003.

[20] J.J. Hopfield and D.W. Tank. Computing with neural circuits: A model. *Science*, 233:625–633, 1986.

[21] B. Warner and M. Misra. Understanding neural networks as statistical tools. *The American Statiscian*, (50:4):284–293, 1996.

[22] C. Calautti and J.C. Baron. Functional neuroimaging studies of motor recovery after stroke in adults - a review. *Stroke*, 34:1553–1566, 2003.

[23] P. Langhorn and F. Coupar A. Pollock. Motor recovery after stroke: a systematic review. *The Lancet Neurology*, 8:741–754, 2009.

[24] C. Stinear. Prediction of recovery of motor function after stroke. *The Lancet Neurology*, 9:1228–1232, 2010.

[25] T.J. Sejnowski and C.R. Rosenberg. Nettalk: a parallel network that learns to read aloud. *The Johns Hopkins University Electrical Engineering and Computer Science Technical Report*, 01:663–672, 1986.

[26] R. Reed. Pruning algorithms - a survey. *IEEE Transactions on Neural Networks*, 4:740–747, 1993.

[27] Y. Le Cun, J.S. Denker, and S.A. Solla. Optimal brain damage. *In D.S. Touretzky (Ed.), Advances in Neural Information Processing Systems*, 2: 598–605, 1990.

[28] B. Hassibi, D.G. Stork, and G.J. Wolff. Optimal brain damage and general network pruning. *1993 IEEE International Conference on Neural Networks, vols 1-3, 293-299*, 1993.

[29] W. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.

[30] F. Rosenblatt. The percetron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:368–408, 1958.

[31] M. Minsky and S. Papert. *Perceptrons*. MIT Press, Cambridge, 1959.

[32] D. L. Eggleton. *Basic Electronics for Scientists and Engineers*. Cambridge University Press, 2011.

[33] P. Werbos. *Beyond regression: New tools for prediction and analysis in the behavioral sciences*. PhD Thesis, Harvard University, 1974.

[34] C.M. Bishop. Neural networks and their applications. *Review of Scientific Instruments*, (65):1803–1832, 1994.

[35] A. Vallido, P.J.G. Lisboa, and J. Vaughan. Neural networks in business: a survey of applications (1992-1998). *Expert Systems with Applications*, (17):51–70, 1999.

[36] M. Paliwal and U.A. Kumar. Neural networks and statistical techniques: A review of applications. *Expert Systems with Applications*, (36):2–17, 2009.

[37] C. M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1995.

[38] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, October 2006.

[39] R. Fletcher. *Practical Methods of Opimization*. Wiley, 1987.

[40] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematical of Control Signals and Systems*, (2):303–314, 1989.

[41] MARINTEK. *Simo theory manual*. Norway, Trondheim, Sintef, 2009.

[42] MARINTEK. *Riflex theory manual*. Norway, Trondheim, Sintef, 2008.

[43] Steen Krenk. *Non-linear Modeling and Analysis of Solids and Structures*. Cambridge University Press, August 2009.

[44] R Capitão and R. Burrows. Wave prediction based on scatter diagram data - a review. *Advances in Engineering Software*, (23):37–47, 1995.

[45] API. *API RP 17G, Recommended Practice for Completion/Workover Risers, Second Editio*. American Petroleum Institute, July 2006 (Reaffirmed April 2007).

[46] Kevin L. Priddy. *Artificial Neural Networks - An Introduction*. SPIE - The International Society for Optical Engineering, 2005.

[47] S.D. Dowling and D.F. Socie. Simple rainflow counting algoritms. *International Journal of Fatigue*, 4:31–40, 1982.

Paper[P1]

Efficient Mooring Line Fatigue Analysis Using a Hybrid Method Time Domain Simulation Scheme

N. H. Christiansen, P. E. T. Voie, J. Høgsberg, N. Sødahl

Proceedings of the OMAE 2013 International Conference on Ocean, Offshore and Arctic Engineering

Proceedings of the OMAE 2013 International Conference on Ocean, Offshore and Arctic Engineering
OMAE 2013
June 9-14, 2013, Nantes, France

OMAE2013-10682

EFFICIENT MOORING LINE FATIGUE ANALYSIS USING A HYBRID METHOD TIME DOMAIN SIMULATION SCHEME

Niels Hørbye Christiansen*
Structures, Pipeline and Risk
DNV Danmark
2900 Hellerup
Denmark
Email: Niels.Christiansen@dnv.com

Per Erlend Torbergsen Voie
Technical Advisory
Det Norske Veritas
7496 Trondheim
Norway
Email: Per.Erlend.Voie@dnv.com

Jan Høgsberg
Department of Mechanical Engineering
Technical University of Denmark
2800 Kgs. Lyngby
Denmark
Email: jhg@mek.dtu.dk

Nils Sødahl
Riser Technology
Det Norske Veritas
1322 Høvik
Norway
Email: Nils.Sodahl@dnv.com

ABSTRACT

Dynamic analyses of mooring line systems are computationally expensive. Over the last decades an extensive variety of methods to reduce this computational cost have been suggested. One method that has shown promising preliminary results is a hybrid method which combines finite element analysis and artificial neural networks (ANN). The present study presents a novel strategy for selecting, arranging and normalizing training data for an ANN. With this approach one ANN can be trained to perform high speed dynamic response prediction for all fatigue relevant sea states and cover both wave frequency motion and slow drift motion. The method is tested on a mooring line system of a floating offshore platform. After training a full fatigue analysis is carried out. The results show that the ANN with high precision provides top tension force histories two orders of magnitude faster than a full dynamic analysis.

*Address all correspondence to this author.

INTRODUCTION

The ever increasing energy demand drives oil and gas exploration towards more and more harsh environments. One field which draws vast attention these years is deep water exploration. Deep water oil exploration leads to an increasing need for structural ability and reliability. Consequences of structural failure in deep water fields are in general very high in terms of environmental, human and economical impact. In order to design safe and reliable structures a lot of attention is put into setting up realistic and sophisticated models for numerical analyzes. The finite element method (FEM) is an important tool in this respect. Time domain simulation of nonlinear systems using FEM analysis can be computationally very expensive - especially when long response histories are needed in order to obtain reliable time series statistics. In design and analysis of flexible risers and mooring systems for floating offshore units, there is a very pronounced need for long term time series simulations - both in evaluation of ultimate limit state and of long term fatigue [1, 2]. Slender offshore structures such as oil/gas risers

and mooring systems exhibit large deformations and therefore require non-linear models for reliable analysis. The combination of the need for detailed non-linear models and the demand for long time series simulations makes design and analysis of these types of structures very time consuming and costly. Thus, the motivation for developing time saving models and methods is very pronounced. As a consequence of this a broad variety of approaches and methods have been suggested over the last decades.

One branch of time saving approaches deals with different types of linearization methods. Many of these methods are based on the Lagrange and updated Lagrange formulations as described by Bathe at el. [3] or Zielinski & Frey [4]. Recently, Kordkheili at el. [5] used a modified updated Lagrange formulation for three dimensional FEM analysis of flexible riser structures. By decomposing strains and stresses Kordkheili at el. obtained a linearization scheme that avoids some of the inaccuracies normally associated with linearization methods when compared to full nonlinear solutions. In analysis of offshore structures there is great desire and tradition for using frequency domain approaches based on wind and wave statistics. But even though extreme response for FPSO (Floating, Production, Storage and Offloading) units and risers/mooring lines often are associated with extreme wave heights and can emerge from resonant situations, as discussed by A.O. Vázquez-Hernández at el. in [6], their nonlinear nature makes full frequency domain analysis inadequate. Spanos at el. [7] compares a statistical linearization technique and coupled time domain analysis for modeling of nonlinear spar dynamics. They show a reliable agreement between the two methods in terms of response statistics and power spectra and suggests that the statistical linearization technique is adequate for initial stages of design analyzes.

In resent years different hybrid methods combining FEM with other methods have shown promising results in dramatic reduction in calculation time. Low [8] has shown how a time/frequency domain hybrid method can reduce computational costs in riser fatigue analysis dramatically. Low combines the non-linear time domain simulation with a frequency domain method separating vibration into wave frequency response and low frequency response. This methods is based on the assumption that wave frequency induced vibrations are approximately linear and therefore can be analyzed by use of a frequency domain approach, while the low frequency response that governs the large deformation is simulated by nonlinear numerical analysis. Another hybrid method philosophy which seems to be booming these years combines classical deterministic models with various types of machine learning techniques or pattern recognition tools. Gupta [9] applies artificial neural networks and principal component analysis in detection of fatigue damage by using the two tools as out layer detectors on observed data sequences. Javier et. al. [10] uses a similar approach to detect damages in bridge-like structures. Among the applied machine learning techniques the application of Artificial Neural Networks (ANN) seems to be the mostly used and the one with the broadest range of applicability. Most of the techniques are based on hybrid methods combining finite element models and ANNs. Ordaz-Hernandez et. al. [11] demonstrate how ANN can be trained to predict the deflection of a nonlinear cantilevered beam. Waszczyszyn & Zieminański [12] demonstrates different possible applications of neural networks in mechanical engineering. In dynamic analysis Xu et. al. [13] identifies structural parameters of a five story building using a neural network on a pre-generated dynamic time domain response history. Guarize et. al. [14] used a similar network structure to simulate the dynamic response of a flexible oil pipe in service and thereby reduced calculation time by a factor of about 20 compared to ordinary nonlinear numerical analysis.

The idea with artificial neural networks is to replicate the human brain's remarkable ability to learn, recognize and predict patterns of different kinds. In the present case the ANN is trained to predict the pattern of motion of a specific structure. This means that in order to learn this pattern the ANN needs an example that shows a motion history corresponding to a given load history. Hence, the ANN cannot stand alone but depends on pre-generated training data. These training data can come from experiments, measurements on real structures or from numerical models such as FEM models. The following study uses training data generated by a FEM model of a structure. FEM models of nonlinear structures often require fine element mesh discretization, small time step size and iterative procedures in order to obtain equilibrium between internal and external forces. For large complex models this can be very time consuming. The ANN's ability to perform nonlinear mapping between a given input and a system output makes it capable of response prediction without equilibrium iterations. Hence, a properly trained ANN can save a lot of computational effort in response prediction. This paper presents a hybrid method approach that can be used in dynamic analysis by predicting the top tension force history for a mooring chain on a floating offshore platform and how this can be used to perform efficient fatigue analyzes. A full fatigue analysis of the mooring lines on the platform is performed based on simulations generated by the optimized ANN. The calculated estimated expected fatigue life of the mooring line system is compared to the result of a equivalent analysis conducted by the two commercial programs for numerical dynamic simulation of marine structures SIMO [15] and RIFLEX [16].

1 Artificial Neural Network

The great advantage of the present hybrid method scheme is that it attempts to take the best from two methods and thereby by-

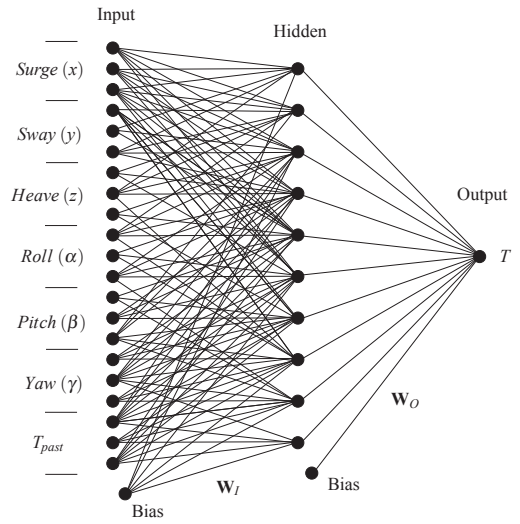


FIGURE 1. Sketch of artificial neural network for predicting top tension force in mooring line.

passes the infuriating compromise between model sophistication and required computational cost. It is well known that complex nonlinear numerical models of large structures are computational demanding. But since the ANN is capable of performing nonlinear mapping between a given input and a corresponding output with great accuracy and without equilibrium iterations, it is possible with a hybrid method to maintain all features of a refined model, e.g. to cover both wave frequency motion and low frequency motion, and still perform fast time domain simulation.

The artificial neural network is a pattern recognition tool that replicates the ability of the human brain to recognize and predict various kinds of patterns. In the following an ANN will be trained to recognize and predict the relationship between loads on a floating platform and the resulting tension forces in a selected anchor chain.

The architecture of a typical one layer artificial neural network is shown in Figure 1. The ANN consists of an input layer, a hidden layer and an output layer. Each connection between two neurons in two neighboring layers has a weight. Training of an ANN corresponds to optimization of these weights for a given data training set. If we want the ANN to completely take over the simulation from the numerical model the network input must be the same as those used for the numerical model in the first place. The number of network outputs can be chosen as one wishes. In principal all output from the numerical model can be included in the ANN. However, as the purpose of the simulation is to determine stresses at one specific location in the anchor chain we limit the output layer to include the response at that specific location only. Both input and output vectors will be described in detail in section 4.

Following [17] the ANN set up and training procedure can be written as follows. The ANN output is calculated by

$$\mathbf{y} = \mathbf{W}_O^\top \mathbf{z}, \qquad \mathbf{z} = \tanh\left(\mathbf{W}_I^\top \mathbf{x}\right), \qquad x_0 \equiv z_0 \equiv 1, \tag{1}$$

where $\mathbf{x}$ is input vector, $\mathbf{y}$ is output vector and $\mathbf{W}_I$ and $\mathbf{W}_O$ are neuron connection weights between input and hidden layer and hidden and output layer, respectively. The tangent hyperbolic is used as activation function between input and hidden layer. The tangent hyperbolic is often used in networks like this, which represent smooth mapping between continuous variables, as it has shown to give rise to fast convergence in network training [18].

Training the ANN correspond to minimizing an error function. The literature suggests many choices of error functions. The simplest and most often used is the sum-of-squares error function which can be written as

$$E(\mathbf{W}) = \frac{1}{2}\sum_{n=1}^{N}\{y(\mathbf{x}_n;\mathbf{W}) - \hat{y}_n\}^2, \tag{2}$$

where y is the ANN output, $\hat{y}$ is the target value and N is the number of samples in the training data set. Optimal weights are found with an iterative procedure stepping in weight space towards minimal error. The weight update is done by gradient decent going the opposite direction of the cost functions gradient as

$$\mathbf{W}_{new} = \mathbf{W}_{old} + \Delta\mathbf{W}, \qquad \Delta\mathbf{W} = -\eta\frac{\partial E(\mathbf{W})}{\partial \mathbf{W}}, \tag{3}$$

where η is the learning step size parameter. This parameter can either be constant or updated during the training of the ANN. For this application, in order to obtain faster convergence, a dynamic learning step size parameter is adjusted for each iteration so that it is increased if the training error is decreased compared to previous iteration step and reduced if the training error increases.

As seen in (3) the training procedure includes the first derivative of the error function with respect to the network weights. This yields

$$\Delta w_{ji} = -\eta\left(\sum_n (1-z_j^2) w_j (y-t) x_i\right) \tag{4}$$

$$\Delta w_j = -\eta\left(\sum_n (y-t) z_j\right) \tag{5}$$

as weight updates for input weights and output weights, respectively. Here i and j correspond to the individual elements in the input vector $\mathbf{x}$ and the hidden layer $\mathbf{z}$.

After training the ANN is ready to take over the time domain simulation from the numerical model. An ANN of this kind with sufficient number of neurons in the hidden layer can be trained to a perfect fit to any training data [18]. However, since the purpose of the training is not to memorize a set of training data but to 'teach' the ANN the underlaying nature of the structure in question it is important that not all of the pregenerated data are use for training but that some part is saved for validation of the ANN.

2 Mechanical Model of Floating Platfrom

To illustrate the hybrid method scheme a model of a mechanical structure is set up. The structure consists of a floating offshore platform at 105 m water depth anchored by 18 mooring lines distributed in 4 clusters, see sketch in Figure 2. The external forces acting on the structure are induced by waves, current and wind.

In principal the dynamic analysis of the platform-mooring system corresponds to solving

$$\mathbf{M}(\mathbf{r})\ddot{\mathbf{r}} + \mathbf{C}(\mathbf{r})\dot{\mathbf{r}} + \mathbf{K}(\mathbf{r})\mathbf{r} = \mathbf{f}(t). \tag{6}$$

where $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{K}$ represent the system inertia, damping and stiffness, respectively. The system inertia matrix account for structural inertia and response dependent hydrodynamic added mass. Linear and nonlinear energy dissipation from internal structural damping and hydrodynamic damping are accounted for by the damping matrix. The stiffness matrix accounts for elastic stiffness and response dependent geometric stiffness. The response of the system is denoted by $\mathbf{r}$ and $\mathbf{f}$ includes all external forces: gravity, buoyancy, hydrodynamic loads etc., acting on the structure.

This means that the platform motion and mooring line tension are mutually dependent. To simplify things the response problem is decoupled. First the platform motions are calculated using a simple quasi-staic catenary mooring line model. Finally, the dynamic mooring line tension is more accurately calculated using a nonlinear finite element mooring line model with prescribed platform motion derived from the precalculated platform motions. Platform motion calculations are carried out by the program SIMO and the dynamic response analysis of the highest stressed line, see Figure 3, is carried out by the program RIFLEX.

Data generated in RIFLEX is used as basis for the hybrid method ANN described in the following sections.

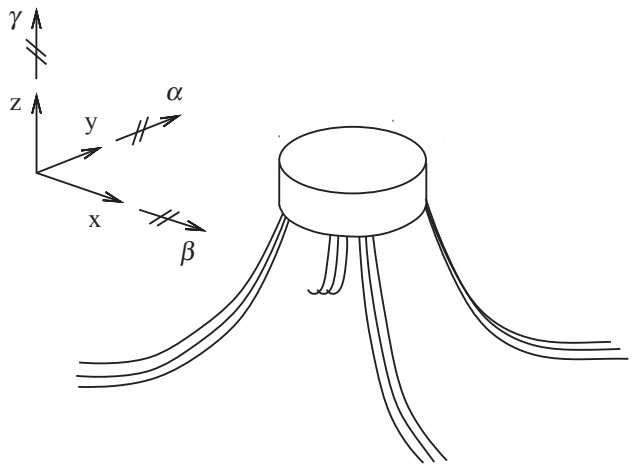


FIGURE 2. FLOATING PLATFORM AND MOORING SYSTEM.

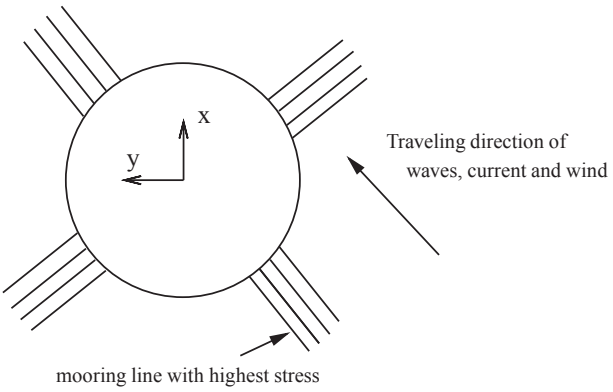


FIGURE 3. FLOATING PLATFORM AND MOORING LINE SYSTEM (TOP VIEW).

3 Selection and Normalization of Training Data

Since the ANN is a tool for pattern recognition it can in principal only predict patterns similar to those used for training of the network. Due to the need for many time series realizations with different wave conditions in fatigue analysis the ANN must be trained to cover a broad range of wave characteristics. This means that the training data must be selected such that all extremes are included but at the same time comprise enough of data in between the extremes as to secure a satisfactory representation of different levels of the nonlinear behavior. The sea states used in training in this example are shown in Table 1. In the table 'X' represent fatigue relevant sea states and 'O' represent sea states selected for network training. Note that the sea state with $H_s = 15$ and $T_p = 17.5$ is not part of the fatigue calculations as it was not part of the original scatter diagram. This sea state

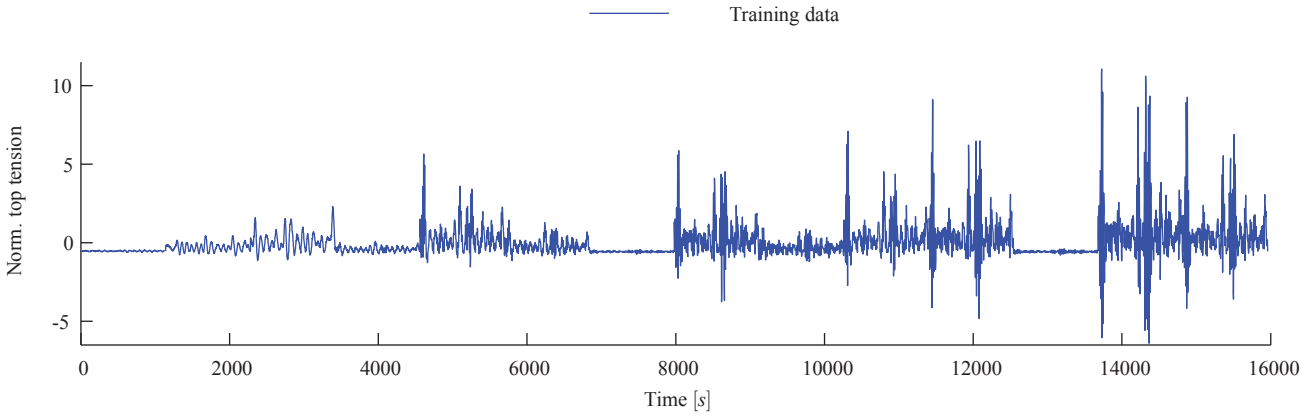


FIGURE 4. TRAINING TARGET DATA BASED ON 15 DIFFERENT SEA STATES.

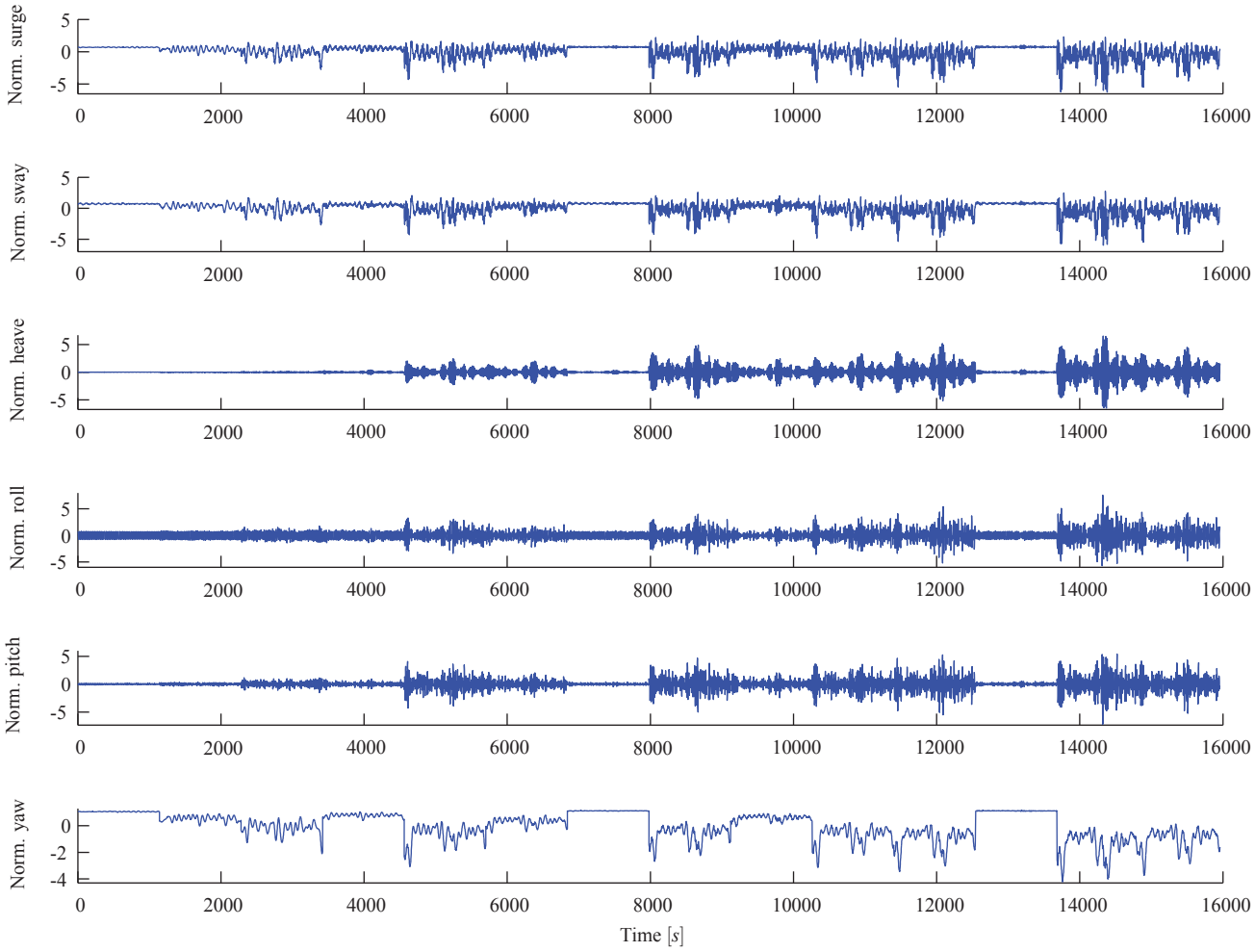


FIGURE 5. TRAINING INPUT DATA BASED ON 15 DIFFERENT SEA STATES.

TABLE 1. SCATTER DIAGRAM OF RELEVANT SEA STATES. TRAINING DATA ARE MARKED BY 'O'.

H_s/T_p	1.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5	10.5	11.5	12.5	13.5	14.5	15.5	16.5	17.5	18.5	19.5	20.5	21.5	22.5
1	X	X	X	X	X	X	X	X	X	X	X	X	O	X	X	X	X	X	O		
2	O	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
4			X	X	X	X	X	X	O	X	X	X	X	X	X	X	X	X			
5				O	X	X	X	X	X	X	X	X	X	X	X	X	X				
6					X	X	X	X	X	X	X	X	X	X	X	O					
7						X	X	X	X	X	X	O	X	X	X						
8							O	X	X	X	X	X	X	X	X						
9								X	X	X	X	X	X	X	X						
10									X	X	X	X	X	X	X						
11										X	X	X	X	O	X						
12											O	X	O	X	O						
13												X	O	O							
15																O					

is merely included to stretch the range of the ANN in order to improve the accuracy for the largest relevant wave heights.

The selected training data are assembled as one long sequence and normalized in the following way.

$$\mathbf{X}_{ANN} = \frac{\mathbf{X}_{FEM} - \mu(\mathbf{X}_{FEM})}{\sigma(\mathbf{X}_{FEM})}, \quad \hat{\mathbf{y}} = \frac{\mathbf{T}_{FEM} - \mu(\mathbf{T}_{FEM})}{\sigma(\mathbf{T}_{FEM})} \tag{7}$$

Normalization of data provides fast training and better convergence as described in most text books on neural networks, eg. [19]. After normalization the data has mean value of $\mu(\mathbf{X}_{ANN}) = 0$ and a standard deviation of $\sigma(\mathbf{X}_{ANN}) = 1$. The same normalization applies for the input that goes into the ANN after training in the subsequent simulation.

The data that is used for training the ANN is shown in Figures 4 and 5. Figure 4 shows the normalized top tension forces for a series containing the 15 different sea states depicted in Table 1. Figure 5 shows normalized histories of the corresponding prescribed six platform motions components as described in section 2.

4 ANN training and optimization

The ANN is designed and trained to predict the top tension of the mooring line. Platform motion is represented by the following six degrees of freedom: surge, sway, heave, roll, pitch and yaw. They are used as ANN input together with the mooring line top tension at the previous time step T_{past} as indicated in Figure 1. This means that the input vector $\mathbf{x}$ at time step t is given by:

$$\begin{aligned}\mathbf{x}_t = \Big[& [\bar{x}(t) \quad \bar{x}(t-1)\dots\bar{x}(t-d)] \quad [\bar{y}(t) \quad \bar{y}(t-1)\dots\bar{y}(t-d)] \\ & [\bar{z}(t) \quad \bar{z}(t-1)\dots\bar{z}(t-d)] \quad [\bar{\alpha}(t) \quad \bar{\alpha}(t-1)\dots\bar{\alpha}(t-d)] \\ & [\bar{\beta}(t) \quad \bar{\beta}(t-1)\dots\bar{\beta}(t-d)] \quad [\bar{\gamma}(t) \quad \bar{\gamma}(t-1)\dots\bar{\gamma}(t-d)] \\ & [\bar{T}(t-1) \quad \bar{T}(t-2)\dots\bar{T}(t-d)]\Big]^T, \end{aligned} \tag{8}$$

where $\bar{x}$, $\bar{y}$, $\bar{z}$, $\bar{\alpha}$, $\bar{\beta}$ and $\bar{\gamma}$ are normalized prescribed platfrom motions, $\bar{T}$ is the is the normalized tension forces from previous time steps and d is number of previous time steps included in the input i.e. the model memory. ANN output is the current mooring line tension in the top chain link at the platform.

$$\mathbf{y}_t = [\bar{T}(t)] \tag{9}$$

Note that in order to obtain the actual tension force $\bar{T}$ must be transformed back using $\mu(\mathbf{T}_{FEM})$ and $\sigma(\mathbf{T}_{FEM})$ calculated during normalization of the training data (7).

In order to obtain an optimal ANN structure three variables are investigated. The ANN is optimized with respect to number

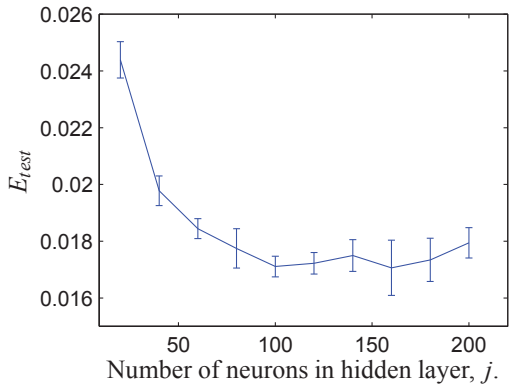


FIGURE 6. OPTIMAL SIZE OF HIDDEN LAYER.

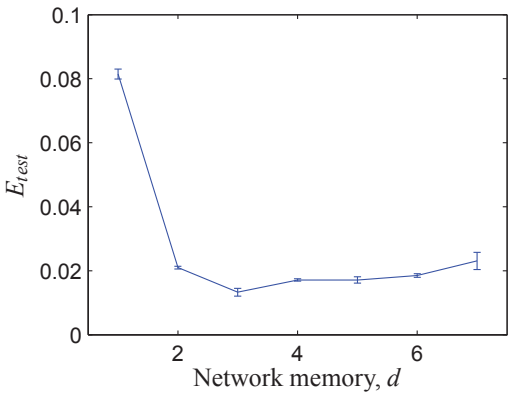


FIGURE 7. OPTIMAL ANN MEMORY.

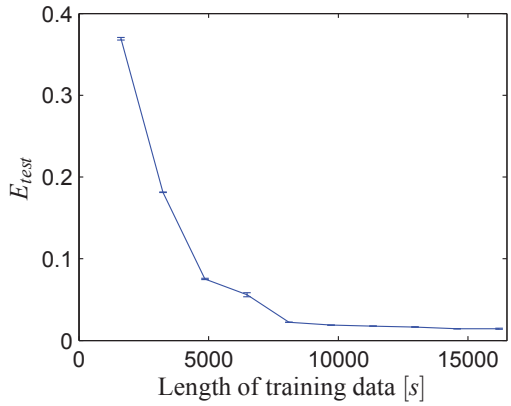


FIGURE 8. OPTIMAL AMOUNT OF TRAINING DATA.

of neurons in the hidden layer, model memory and amount of training data. As mentioned earlier the simulation data generated by the RIFLEX model is divided into two sets - a training set and a validation set. The purpose of the training set is to teach the ANN the nature of the structure i.e. the underlaying mapping between RIFLEX input and output. The validation set is used to test the performance of the trained ANN with respect to fresh data. The validation error (or test error) is calculated in the same way as the training error (2). This validation error must be monitored during training of the ANN. The reason is that even though the objective of the training is to minimize the training error, the ultimate goal is to get the validation error as low as possible.

In Figure 6 the validation error E_{test} is plotted against the number of neurons in the hidden layer. The figure shows the results of five runs. The line indicates the mean value of the error of the validation set over the five runs. The vertical bars indicate the standard deviation. The figure shows that the error and the scatter in performance of the trained ANN reach a minimum when the number of neurons is about 100. Apparently the figure also shows that there is no gain in increasing the number of neurons beyond 120. In fact the opposite seems to be the case, as the mean error and scatter increase slightly as the number of hidden units grow. Therefore, 100 neurons in the hidden layer are used in the following calculations.

In Figure 7 the validation error is plotted against the model memory - that is how many previous time steps that are used as network input - corresponding to d in (8). The picture shows that including memory in the model reduces the error significantly. On the other hand it is seen that increasing the memory to include more than three steps does not improve the ANN performance. In fact there seems to be a minimum validation error a $d = 3$. Thus, a memory of three time steps is used in the following.

Figure 8 shows the validation error against amount of training data. It is clear that selecting a sufficient amount of training data is vital in order for the ANN to perform satisfactorily. As seen in Figure 5 a total of 1.6e4 seconds simulated data is used for ANN training in this example. Note that these set of data covers all 15 selected sea states. The amount of required training data obviously depend on the number of included sea states.

With 7 input tracks (6 prescribed motion components and 1 from previous tension force) and a 3 time step memory the input vector $\mathbf{x}$ for the ANN will have the dimension $[22 \times 1]$. And with 100 neurons in the hidden layer the input weight matrix $\mathbf{W}_I$ has the dimension $[100 \times 22]$. As the ANN has just one output the output matrix $\mathbf{W}_O$ reduces to a vector of size $[101 \times 1]$

In order to keep the underlaying equilibrium algorithm in RIFLEX stable the time step of the dynamic simulation is 0.1 s. For the ANN, the time step size must be chosen so that the network is able to recognize the dynamics of the structure. Therefore, in many cases the ANN is capable of handling a larger time step size than the corresponding numerical model. Using a larger time step for the ANN, i.e. omitting a number of data points, means that it is possible to cover more vibration cycles with the same training data sample size and hence reduce the time spend on ANN training. For this particular structure a time step of 2

seconds have shown to give good results. Thus, 95 % of the data can be ignored without compromising the accuracy of the ANN. This is, however, highly dependent on the response frequencies of the structure why the time step size of 2 seconds can not be considered as a general appropriate step size.

5 Simulation using the optimized ANN

When the ANN is trained it is able to generate nonlinear output without equilibrium iterations and hence at a remarkable pace. Figure 9 shows comparisons of RIFLEX and ANN simulations of the top tension force in the mooring line for three different randomly picked sea states (sea states that where not part of the ANN training). For this particular simulation the trained ANN calculates a factor of about 600 times faster than RIFLEX. However, one must still consider the total costs of the hybrid method when assessing wether the hybrid method is cost effective.

1. Setting up the artificial neural network
2. Selecting training data
3. Training the network

As it is stated in many textbooks on ANNs, obtaining an optimal network is more an art form than a well defined science. Hence, finding a reliable network structure is a matter of trial and error and requires some degree of experience. However, setting up the network probably must only be done once. After finding an appropriate network architecture the same network will then be able to handle most slender marine structures.

If this method shall be useful on a larger scale, it is vital to have a robust standard procedure for selecting sea states used for training. As mentioned earlier, the training data must cover the entire desired range of the ANN - at least so that extremes and some intermediate levels are included. In a case like this where different sea states do not contribute equally to the fatigue damage of the structure. Some types of waves contribute a lot to the fatigue because they cause large stresses in the structure (typically large waves and waves with peak periods close to eigenperiods of the structure) while other more moderate waves contribute significantly due to their large number. The training data should be selected so that the main focus is on the most harmful sea states seen over the complete life time. Since this information is part of the result of the analysis one must rely on experience to evaluate this.

The training of the network is a fairly quick task. However, the training must be repeated every time the required simulation moves outside any previous 'training data range'. This means that if something in the analysis is changed and this change influences the response pattern the network must be retrained.

6 Fatigue life evaluation

To evaluate the accuracy of the hybrid method a full life time assessment for the mooring line is carried out. The expected life time is estimated based on results obtained by hybrid method simulation and then compared to the estimate obtained by the full RIFLEX analysis. The analysis is done in accordance with DNV's offshore standard and recommended practice for riser analysis [1, 2].

For the fatigue analyses a complete Hs-Tp scatter diagram is applied. Only the worst weather direction with respect to mooring line fatigue damage is considered. Short term sea states are constructed of the maximum significant wave height and weighted spectral peak period within each block of the scatter diagram and an associated wind speed and an associated current speed. The wind speed associated with the significant wave height is found by matching the directional cumulative distribution function (CDF) of significant wave heights with the directional CDF of wind speeds. Similarly, the current speed associated with the significant wave height is found by matching the omni-directional CDF of significant wave heights with the omni-directional CDF of current speeds. Waves, wind and current are assumed to act collinearly. Waves are simulated from a Torsethaugen wave spectrum, wind speed is simulated from the ISO wind spectrum. Time series of platform motion are simulated for each of the sea states defined above using SIMO. Nonlinear load effect analyses of the mooring line subjected to forced platform motion and wave loads are performed with RIFLEX using platform motions from SIMO. Current loads on the mooring line are not included. Fatigue damage is computed for the mooring chain link located at the fairlead applying the DNV studless chain SN-curve and stress concentration factors 1.15.

The damage is calculated by a Palmgren-Miner summation of the stress variation cycles from a rain flow count of each simulation history. The damage from each sea state is then multiplied with the likelihood of the given sea state. This gives the expected annual damage and hence the inverse of the expected fatigue life of the anchor line when summed over all sea states. The results are listed in Table 2. It is seen that the hybrid method deviates by only 1.6 % from the RIFLEX analysis, which in this case must be considered as the correct estimate.

Considering that the hybrid method is more than two orders of magnitude faster than the RIFLEX analysis the 1.6 % deviation is remarkable. If the length of the time domain simulations is increased the reliability of the analysis is also increased. With the RIFLEX model increasing simulation lengths are quite expensive, whereas with the hybrid method the simulation length does not make any practical difference as the simulation it self after training is so fast (about 600 times faster than RIFLEX).

Figures 10 and 11 show the accumulated damage for all significant wave heights and wave peak periods, respectively. It is seen that the ANN tends to overestimate the damage on the smallest wave heights whereas the damage from the large wave

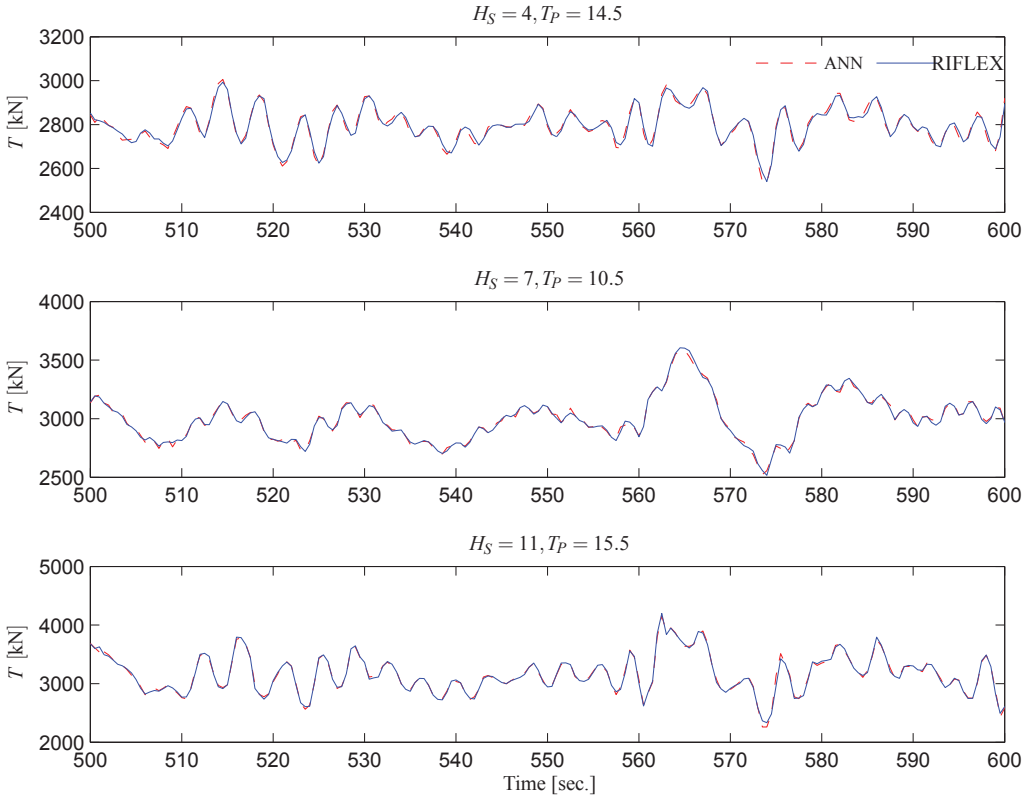


FIGURE 9. ANN AND RIFLEX SIMULATION FOR TIME SERIES FOR 3 DIFFERENT SEA STATES.

TABLE 2. Calculated expected fatigue life for the anchor line.

	Fatigue life [years]	Deviation [%]
RIFLEX	1698	-
Hybrid	1725	~ 1.6 %

height is underestimated. The same tendency is seen for wave periods. One could argue that the 1.6 % deviation is a matter of luck since the over- and underestimations coincidentally cancels each other out. However the figures clearly show that the ANN simulations are very close to the RIFLEX simulations on all individual levels.

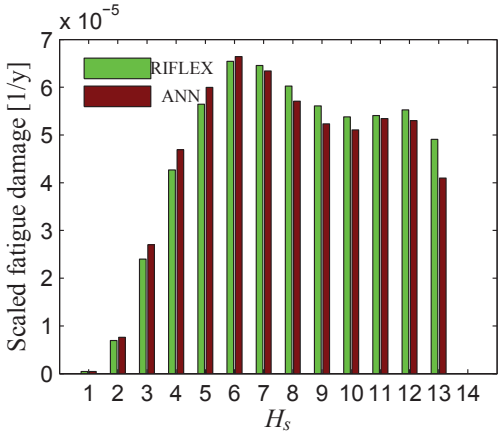


FIGURE 10. ACCUMULATED YEARLY DAMAGE H_S.

7 Concluding remarks

In the example presented in this paper the reduction in CPU time spend on a simulation of the mooring line top tension force history is about a factor of 600 when using the ANN compared to RIFELX - that is when the ANN is set up and trained. This factor is obviously dependent on the structure, the loading, model configuration etc. However, the results presented in this paper indicate that the hybrid method holds a great potential.

Since the ANN relies on training data, and hence a model to generate these data, e.g. a FEM model, the hybrid method is

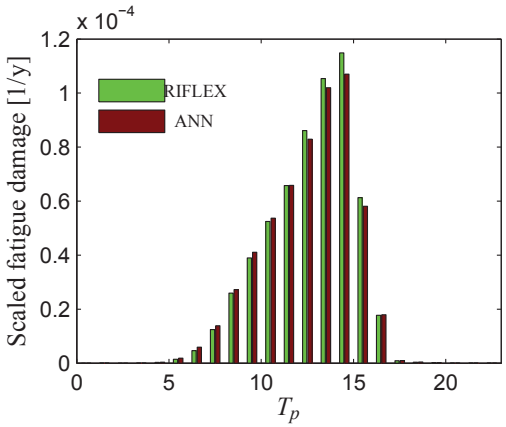


FIGURE 11. ACCUMULATED YEARLY DAMAGE T_P.

only relevant in cases where several long time series simulations are needed. However, this is exactly the case for most nonlinear structures subject to large displacements and fatigue.

It is shown how a fatigue calculation that usually takes 10+ hours, with this hybrid method can be completed within an hour. However, designers using this method must be very careful when selecting training data. It is important that the training data covers the whole range of the final simulated data and also that the training set contains a sufficient amount of data. In order for the hybrid method to be useful on a larger scale a reliable training data selection procedure must be standardized. This will require experience and methods to predict the most damaging sea states and required amount of training data before an analysis is performed. Furthermore, due to the 'black box' nature of the ANN it is vital to perform a reliable and adequate validation of the trained ANN before conducting a full analysis.

REFERENCES

[1] DNV, 2010. *Offshore Standard DNV-OS-F201 - Dynamic Risers*. Det Norske Veritas, October.

[2] DNV, 2005. *Recommended Practice DNV-RP-F204 - Riser Fatigue*. Det Norske Veritas, July.

[3] Bathe, K. J., Ramm, E., and Wilson, E. L., 1975. "Finite element formulations for large deformation dynamic analysis". *International Journal for Numerical Methods in Engineering,* ***9***, pp. 353–386.

[4] Zieliński, A. P., and Frey, F., 2000. "On linearization in non-linear structural finite element analysis". *Computers and Structures,* ***79***, pp. 825–838.

[5] Kordkheili, S. H., Bahai, H., and Mirtaheri, M., 2011. "An updated lagrangian finite element formulation for large displacement dynamic analysis of three-dimensional flexible riser structures". *Ocean Engineering,* ***38***, pp. 793–803.

[6] Vázquez-Hernández, A., Ellwanger, G., and Sagrilo, L., 2011. "Long-term response analysis of fpso mooring systems". *Applied Ocean Research,* ***33***, pp. 375–383.

[7] Spanos, P. D., Nava, V., and Arena, F., 2011. "Coupled surge-heave-pitch dynamic modeling of spar-monopol-riser interaction". *Journal of Offshore Mechanics and Arctic Engineering,* ***133***.

[8] Low, Y., 2011. "Extending a time/frequency domain hybrid method for riser fatigue analysis". *Applied Ocean Research,* ***33***, pp. 79–87.

[9] Gupta, S., Ray, A., and Keller, E., 2005. "Symbolic time series analysis of ultrasonic data for early detection of fatigue damage". *Mechanical Systems and Signal Processing,* ***21***, pp. 866–884.

[10] Valentin-Sivico, J., Rao, V. S., and Samanthula, V., 1998. "Damage detection of bridge-like structures using neural networks". *Smart Structures and Materials: Smart Systems for Bridges, Structures, and Highways - Proceedings of SPIE,* ***3325***, pp. 139–148.

[11] K. Ordaz-Hernandez, X. F., and Bennis, F., 2007. "Modal reduction technique for mechanical behaviour modelling: Efficiency criteria and validity domain assessment". *Journal of Mechanical Engineering Science,* ***222***, pp. 493–505.

[12] Waszczyszyn, Z., and Ziemiański, L., 2001. "Neural networks in mechanics of structures and materials - new results and prospects of applications". *Computers and Structures,* ***79***, pp. 2261–2276.

[13] Xu, B., Wu, Z., and Yokoyama, K., 2003. "Neural network based modeling of structural parametric evaluation with dynamic respones in time domain". *Smart Nondestructive Evaluation and Health Monitoring of Structural and Biological Systems II - Proceedings of SPIE,* ***5047***, pp. 392–402.

[14] Guarize, R., Matos, N., Sagrilo, L., and Lima, E., 2008. "Neural networks in dynamic response analysis of slender marine structures". *Applied Ocean Research,* ***29***, pp. 191–198.

[15] MARINTEK, 2009. *Simo theory manual*. Norway, Trondheim, Sintef.

[16] MARINTEK, 2008. *Riflex theory manual*. Norway, Trondheim, Sintef.

[17] Bishop, C. M., 2006. *Pattern Recognition and Machine Learning*. Springer, New York.

[18] Bishop, C. M., 1995. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford.

[19] Priddy, K. L., 2005. *Artificial Neural Networks - An Introduction*. SPIE - The International Society for Optical Engineering.

Paper[P2]

Comparison of Neural Network Error Measures for Simulation of Slender Marine Structures

N. H. Christiansen, P. E. T. Voie, O. Winther, J. Høgsberg

Journal of Applied Mathematics, (2014)

Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2014, Article ID 759834, 11 pages
http://dx.doi.org/10.1155/2014/759834

Research Article

Comparison of Neural Network Error Measures for Simulation of Slender Marine Structures

Niels H. Christiansen,[1,2] Per Erlend Torbergsen Voie,[3] Ole Winther,[4] and Jan Høgsberg[2]

[1] *DNV Denmark A/S, Tuborg Parkvej 8, 2900 Hellerup, Denmark*
[2] *Department of Mechanical Engineering, Technical University of Denmark, 2800 Kongens Lyngby, Denmark*
[3] *Det Norske Veritas, 7496 Trondheim, Norway*
[4] *Department of Applied Mathematics and Computer Science, Technical University of Denmark, 2800 Kongens Lyngby, Denmark*

Correspondence should be addressed to Niels H. Christiansen; niels.christiansen@dnv.com

Received 19 November 2013; Accepted 21 January 2014; Published 2 March 2014

Academic Editor: Ricardo Perera

Training of an artificial neural network (ANN) adjusts the internal weights of the network in order to minimize a predefined error measure. This error measure is given by an error function. Several different error functions are suggested in the literature. However, the far most common measure for regression is the mean square error. This paper looks into the possibility of improving the performance of neural networks by selecting or defining error functions that are tailor-made for a specific objective. A neural network trained to simulate tension forces in an anchor chain on a floating offshore platform is designed and tested. The purpose of setting up the network is to reduce calculation time in a fatigue life analysis. Therefore, the networks trained on different error functions are compared with respect to accuracy of rain flow counts of stress cycles over a number of time series simulations. It is shown that adjusting the error function to perform significantly better on a specific problem is possible. On the other hand. it is also shown that weighted error functions actually can impair the performance of an ANN.

1. Introduction

Over the years, oil and gas exploration has moved towards more and more harsh environments. In deep and ultra-deep water installations the reliability of flexible risers and anchor lines is of paramount importance. Therefore, the design and analysis of these structures draw an increasing amount of attention.

Flexible risers and mooring line systems exhibit large deflections in service. Analysis of this behavior requires large nonlinear numerical models and long time-domain simulations [1, 2]. Thus, reliable analysis of these types of structures is computationally expensive. Over the last decades an extensive variety of techniques and methods to reduce this computational cost have been suggested. A review of the most common concepts of analysis is given in [3]. One method that has shown promising preliminary results is a hybrid method which combines the finite element method (FEM) and artificial neural network (ANN) [4]. The ANN is a pattern recognition tool that based on sufficient training can perform nonlinear mapping between a given input and a corresponding output. The reader may consult, for example, Warner and Misra [5] for a fast thorough introduction to neural networks and their features. The idea with the hybrid method is to first perform a response analysis for a structure using a FEM model and then subsequently to use these results to train an ANN to recognize and predict the response for future loads. As demonstrated by Ordaz-Hernandez et al. [6] an ANN can be trained to predict the deformation of a nonlinear cantilevered beam. A similar approach was used by Hosseini and Abbas [7] when they predicted the deflection of clamped beams struck by a mass. In connection with analysis of marine structures Guarize et al. [8] have shown that a well-trained ANN with very high accuracy can reduce dynamic simulation time for analysis of flexible risers by a factor of about 20. However, a problem with this method is that an ANN can only make accurate predictions

based on sufficiently known input patterns. This means that a network trained on one type of load pattern will have difficulties in predicting the response when the structure is exposed to different types of loading conditions. Recently, a novel strategy for arranging the training data has been proposed by Christiansen et al. [9], where the idea is to select small samples of simulated data for different sea states and collect these in one training sequence. With a proper selection of data an ANN can be trained to predict tension forces in a mooring line on a floating offshore platform for a large range of sea states two orders of magnitude faster than the corresponding direct time integration scheme. It has been shown how computation time, when conducting the simulations associated with a full fatigue analysis on a mooring line system on a floating offshore platform, can be reduced from about 10 hours to less than 2 minutes.

Training of an ANN corresponds to minimizing a predefined error function. Several studies on the efficiency of using different objective functions for ANN training have been conducted over the last decades. Accelerated learning in neural networks has been obtained by Solla et al. [10] using relative entropy as the error measure and Hampshire and Waibel [11] presented an objective function called classification figure of merit (CFM) to improve phoneme recognition in neural networks. A comparative study performed by Altincay and Demirekler [12] showed that the CFM objective function also improved the performance of neural networks trained as classifiers for speaker identification. However, these references all consider networks used for classification whereas the one used in this paper is trained to perform regression between time-continuous variables. Hence, the problem studied in this paper calls for different cost functions from those used for neural classifiers.

This study evaluates and compares four different error functions with respect to ANN performance on the fatigue life analysis of the same floating offshore platform as used in [9]. A numerical model of a mooring line system on a floating platform subject to current, wind, and wave forces is established. The model is used to generate several 3-hour time domain simulations at seven different sea states with 2 m, 4 m, ..., and 14 m significant wave height, respectively. The generated data is then divided into a training set and a validation set. The training set consists of series of simulations at only the sea states with 2 m, 8 m, and 14 m wave height. The remaining part is then left for validation of the trained ANN. The full numerical time integration analysis is carried out by the two tailor-made programs SIMO [13] and RIFLEX [14], while the neural network simulations are conducted by a small MATLAB toolbox.

2. Artificial Neural Network

The artificial neural network is a pattern recognition tool that replicates the ability of the human brain to recognize and predict various kinds of patterns. In the following an ANN will be trained to recognize and predict the relationship between the motion of a floating platform and the resulting tension forces in a specific anchor chain.

2.1. Setting Up the ANN. The architecture of a typical one layer artificial neural network is shown in Figure 1. The ANN consists of an input layer, where each input neuron represents a measured time discrete state of the system. In the present case in Figure 1 the neurons of the input layer represent the six motion components (x, y, ...) of the floating platform and the previous time discrete anchor chain tension force (T_{past}). The input layer is connected to a single hidden layer, which is then connected to the output layer representing the tension force (T). Two neurons in neighboring layers are connected and each of these connections has a weight. The training of an ANN corresponds to an optimization of these weights with respect to a particular data training set. The accuracy and efficiency of the network depend on the network architecture, the optimization of the individual weights, and the choice of error function used in the applied optimization procedure.

The design and architecture of the ANN and the subsequent training procedure follow the approach outlined in [15]. Assume that the vectors $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ contain the neuron variables of the input layer, output layer, and hidden layer, respectively. The output layer and hidden layer values can be calculated by the expressions

$$\mathbf{y} = \mathbf{W}_O^{\top}\mathbf{z}, \qquad \mathbf{z} = \tanh\left(\mathbf{W}_I^{\top}\mathbf{x}\right), \qquad x_0 \equiv z_0 \equiv 1, \quad (1)$$

where $\mathbf{W}_I$ and $\mathbf{W}_O$ are arrays that contain the neuron connection weights between the input and the hidden layer and the hidden and the output layer, respectively. By setting x_0 and z_0 permanently to one, biases in the data can be absorbed into the input and hidden layer. The tangent hyperbolic function is used as an activation function between the input and the hidden layer. A nonlinear activation function is needed in order to introduce nonlinearities into the neural network. The tangent hyperbolic is often used in networks of this type, which represent a monotonic mapping between continuous variables, because it provides fast convergence in the network training procedure; see [16].

The optimal weight components of the arrays $\mathbf{W}_I$ and $\mathbf{W}_O$ are found by an iterative procedure, where the weights are modified to give a minimum with respect to a certain error function. The updating of the weight components is performed by a classic gradient decent technique, which adjusts the weights in the opposite direction of the gradient of the error function [17]. For the ANN this gradient decent updating can be written as

$$\mathbf{W}_{\text{new}} = \mathbf{W}_{\text{old}} + \Delta\mathbf{W}, \qquad \Delta\mathbf{W} = -\eta\frac{\partial E(\mathbf{W})}{\partial \mathbf{W}}, \quad (2)$$

where E is a predefined error function and η is the learning step size parameter. This parameter can either be constant or updated during the training of the ANN. For the applications in the present paper the learning step size parameter is dynamic and will be adjusted for each iteration so that it is increased if the training error is decreased compared to previous iteration steps and reduced if the error increases.

2.2. Error Functions. As mentioned above in Section 2.1 the training of an ANN corresponds to minimizing the associated measure of error represented by the predefined error function

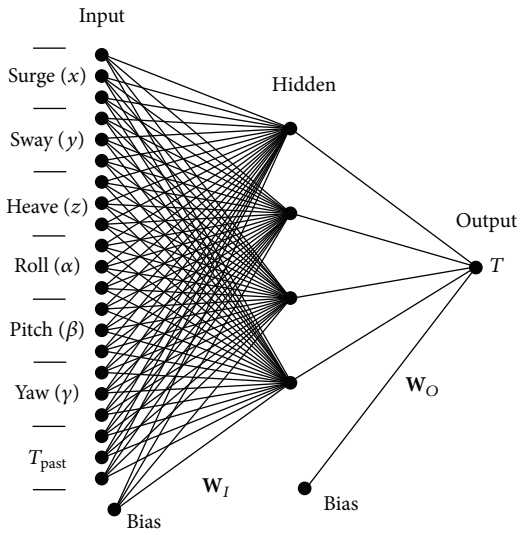


Figure 1: Sketch of artificial neural network for predicting top tension force in mooring line.

E. The literature suggests many choices of error functions [16].

The simplest and most commonly used error function in neural networks used for regression is the mean square error (MSE). However, the purpose of the present ANN is to significantly reduce the calculation time for a fatigue analysis of the marine type structure. And since the large amplitude stresses contribute far the most to the accumulated damage of the mooring lines it is of interest to investigate how a different choice of error measure will affect the accuracy and efficiency of the fatigue life calculations. Four different error functions are therefore tested and compared to the full numerical solution obtained by time simulations using the RIFLEX code.

The comparison is based on the so-called *Minkowski-R* error:

$$E = \frac{1}{R}\sum_{n}\sum_{k=1}^{c}\left|y_k\left(\mathbf{x}_n;\mathbf{W}\right)-t_{kn}\right|^R, \tag{3}$$

where y is the scalar ANN output and t is the target value. The classic MSE is seen to be a special case of the Minkowski error with $R = 2$. In many situations the performance and accuracy of the ANN are equally important regardless of the magnitude of the actual output. However, when dealing with analysis of structures this is not always the case. For example, the purpose of the ANN in the present paper is to simulate the top tension force time history in a mooring line, which is subsequently used to evaluate the fatigue life of the line. And since by far the most damage in the mooring line is introduced by large amplitude stress cycles, the ANN inaccuracy on large stresses is much more expensive than errors on small and basically unimportant amplitudes. One way to specifically emphasize large amplitudes is to increase the R-value in (3). Another and more direct way to place additional focus on the importance of large stress amplitudes is by multiplying each term in (3) by the absolute value of the target values. This yields the following weighted error function:

$$E_w = \frac{1}{R}\sum_{n}\sum_{k=1}^{c}\left|y_k\left(\mathbf{x}_n;\mathbf{W}\right)-t_{kn}\right|^R\cdot\left|t_{kn}\right|^R. \tag{4}$$

The performance of a trained ANN is usually measured in terms of the so-called validation error, which is calculated in the same way as the training error but on entirely new data set that has not been part of the network training. This means that when comparing networks that have been trained using different error functions the validation error is the common measure to assess performance in terms of accuracy and computational effort. Obviously the various networks considered in the following could be tested and compared against any of the error functions that the networks have been trained by. But that would definitely favor the particular ANN that has been trained by the specific error function that is chosen as the validation measure. And since the ultimate objective of the ANN is to predict the fatigue life of the mooring line it is appropriate to calculate and compare the accumulated damage in the mooring line caused by all seven sea state realizations previously mentioned in Section 1.

2.3. Network Training. In (2) the steepest decent correction of the weight vector for the training of the network requires the first derivative of the error function E with respect to the weight arrays $\mathbf{W}$. Differentiation of (3) with respect to the components of the two weight matrices $\mathbf{W}_I$ and $\mathbf{W}_O$ yields

$$\begin{aligned}\frac{dE}{dW_{O,kj}} &= \sum_{n}\left|y_k\left(\mathbf{x}_n;\mathbf{W}\right)-t_{kn}\right|^{R-1}z_j,\\ \frac{dE}{dW_{I,ji}} &= \sum_{n}\left(\left(1-z_j^2\right)\sum_{k=1}^{c}w_{kj}\left|y_k\left(\mathbf{x}_n;\mathbf{W}\right)-t_{kn}\right|^{R-1}\right)x_i.\end{aligned} \tag{5}$$

These gradients are now inserted into (2) and thereby govern the correction of the weights for each iteration step in

the training procedure. Similar differentiation of the weighted error function (4) yields

$$\frac{dE_w}{dW_{O,kj}} = \sum_n \left| y_k\left(\mathbf{x}_n; \mathbf{W}\right) - t_{kn}\right|^{R-1} z_j \cdot \left|t_{kn}\right|,$$

$$\frac{dE_w}{dW_{O,ji}} = \sum_n \left(\left(1 - z_j^2\right) \sum_{k=1}^{c} w_{kj} \left| y_k\left(\mathbf{x}_n; \mathbf{W}\right) - t_{kn}\right|^{R-1} \right) x_i \cdot \left|t_{kn}\right|. \tag{6}$$

The above equations are implemented into the training algorithm and tested for the two power values $R = 2$ and $R = 4$. This gives a total of four different error functions, which in the following will be denoted as

(i) E_2: unweighted error function with $R = 2$;

(ii) E_2^w: weighted error function with $R = 2$;

(iii) E_4: unweighted error function with $R = 4$;

(iv) E_4^w: weighted error function with $R = 4$.

It should be noted that the first case E_2 represents the classic MSE function.

3. Application to Structural Model

The structure used as the basis for the comparison of the different error functions is shown in Figure 2. It consists of a floating offshore platform located at 105 m water depth, which is anchored by 18 mooring lines assembled in four main clusters. The external forces acting on the structure are induced by waves, current, and wind.

3.1. Structural Model. In principal the dynamic analysis of the platform-mooring system corresponds to solving the equation of motion:

$$\mathbf{M}\left(\mathbf{r}\right)\ddot{\mathbf{r}} + \mathbf{C}\left(\mathbf{r}\right)\dot{\mathbf{r}} + \mathbf{K}\left(\mathbf{r}\right)\mathbf{r} = \mathbf{f}\left(t\right). \tag{7}$$

In this nonlinear equation $\mathbf{r}$ contains the degrees of freedom of the structural model, and $\mathbf{f}$ includes all external forces acting on the structure from, for example, gravity, buoyancy, and hydrodynamic effects, while the nonconstant matrices $\mathbf{M}$, $\mathbf{C}$, and $\mathbf{K}$ represent the system inertia, damping, and stiffness, respectively. The system inertia matrix $\mathbf{M}$ takes into account both the structural inertia and the response dependent hydrodynamic added mass. Linear and nonlinear energy dissipation from both internal structural damping and hydrodynamic damping are accounted for by the damping matrix $\mathbf{C}$. Finally, the stiffness matrix contains contributions from both the elastic stiffness and the response dependent geometric stiffness.

The nonlinear equations of motion in (7) couple the structural response of the floating platform and the response of the mooring lines. However, the system is effectively solved by separating the solution procedure into the following steps. First the motion of the floating platform is computed by the program SIMO [13], assuming a quasistatic catenary mooring line model with geometric nonlinearities. The platform response from this initial analysis is subsequently used as excitation in terms of prescribed platform motion in a detailed nonlinear finite element analysis for the specific mooring line with highest tension stresses. The location of the mooring line with largest stresses is indicated in Figure 3. For this specific line the hot-spot with respect to fatigue is located close to the platform and is in the following referred to as the top tension force. From the detailed fully nonlinear analysis performed by RIFLEX the time history of the top tension force at this hot-spot is extracted.

Based on the simulated time histories for both the platform motion and the top tension force an ANN is trained to predict the top tension force in the selected mooring line with the platform response variables as network input. This is considered next in Section 3.2. In [9] a multilayer ANN was trained to simulate the top tension force two orders of magnitude faster than a corresponding numerical model. The training data was set up and arranged so that a single ANN with a single hidden layer could simulate all fatigue relevant sea states and thereby provide a significant reduction in the computational effort associated with a fatigue life evaluation. For clarity the ANN used in this example covers only a few sea states, with different significant wave heights and constant peak period. This gives a compact neural network that is conveniently used to illustrate the influence of changing the error function in the training of the network.

3.2. Selection of Training Data. The ultimate purpose of the ANN is to completely bypass the computationally expensive numerical time integration procedure, which in this case is conducted by the RIFLEX model. This means that the input to the neural network must be identical to the input used for the RIFLEX calculations. In this case the input is therefore the platform motion, represented by the six degrees of freedom denoted in Figure 1 and illustrated in Figure 2. In principle the number of neural network output variables can be chosen freely, and in fact all degrees of freedom from the numerical finite element analysis can be included as output variables in the corresponding ANN. However, the strength of the ANN in this context is that it may provide only the output variable that drives the design of the structure, which in this case is the maximum top tension forces in the particular mooring line. This leads to a very fast simulation procedure, which for a well-trained network provides sufficiently accurate results. Thus, the ANN is in the present case designed and trained to predict the top tension of the mooring line, and the platform motion (six motion components; surge, sway, heave, roll, pitch, and yaw) is, together with the top tension of previous time steps, used as input to the ANN; see Figure 1. This means that the input vector $\mathbf{x}_n$ at time increment n can be constructed as

$$\begin{aligned}\mathbf{x}_n = [\ & \left[x_t \ \ x_{t-h} \ \cdots \ x_{t-dh}\right], \left[y_t \ \ y_{t-h} \ \cdots \ y_{t-dh}\right], \\ & \left[z_t \ \ z_{t-h} \ \cdots \ z_{t-dh}\right], \left[\alpha_t \ \ \alpha_{t-h} \ \cdots \ \alpha_{t-dh}\right], \\ & \left[\beta_t \ \ \beta_{t-h} \ \cdots \ \beta_{t-dh}\right], \left[\gamma_t \ \ \gamma_{t-h} \ \cdots \ \gamma_{t-dh}\right], \\ & \left[T_{t-h} \ \ T_{t-2h} \ \cdots \ T_{t-dh}\right] \ ]^T,\end{aligned} \tag{8}$$

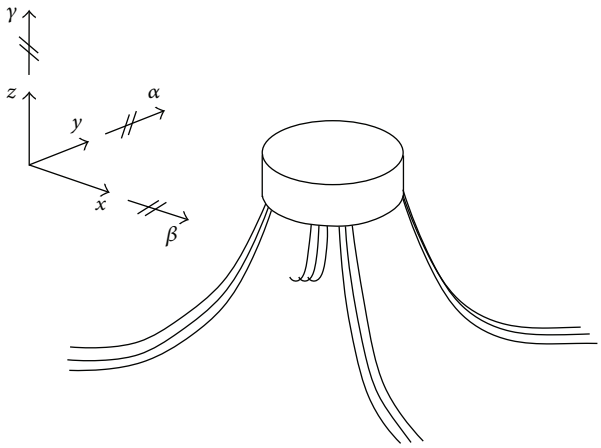


Figure 2: Mooring system with floating platform and anchor lines.

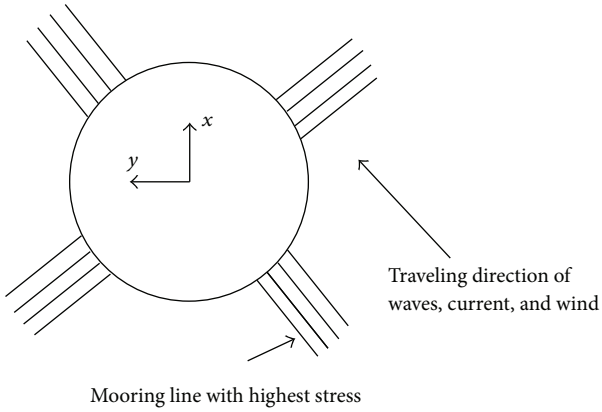


Figure 3: Mooring line system (top view).

where $t = nh$ denotes current time, h is the time increment, and d is the number of previous time steps included in the input, that is, the model memory. The corresponding ANN output is the value of the top tension force T_t in the mooring line:

$$y_n = T_t. \tag{9}$$

Since there is only one network output y is a scalar and not a vector as in (1). For the training of the ANN nonlinear simulations in RIFLEX are conducted for sea states with a significant wave height of H_s = 2 m, 8 m, and 14 m, respectively. While neural networks are very good at interpolating within the training range, they are typically only able to perform limited extrapolation outside the training range. Thus, the selected training set must contain both the minimum wave height (2 m), the maximum wave height (14 m), and in this case a moderate wave height (8 m) to provide sufficient training data over the full range of interest. With these wave heights included in the training data the ANN is expected to be able to provide accurate time histories for the top tension force for all intermediate wave heights. The seven 3-hour simulation records generated by RIFLEX are divided into a training set and a validation set. The data that is used for training of the ANN is shown in Figures 4 and 5. Figure 4 shows the time histories for the six motion degrees of freedom of the platform calculated by the initial analysis in SIMO and used as input to both the full numerical analysis in RIFLEX and the ANN training and simulation. Figure 5 shows corresponding time histories for top tension force determined by RIFLEX. The full time histories shown in Figures 4 and 5 are constructed of time series for the three significant wave heights. The first 830 seconds of the training set represent 2 m significant wave height, the next 830 seconds are for 8 m wave height, and the remaining part is then for 14 m wave height.

The SIMO simulations are conducted with a time step size of 0.5 s. In the subsequent RIFLEX simulations the time step must be sufficiently small as to keep the associated Newton-Raphson iteration algorithm stable. In these simulations a time step size of 0.1 s is therefore chosen, which means that the additional input parameters are obtained by linear interpolation between the simulation values from SIMO. For the ANN the time step size h must be chosen so that the network is able to grasp the dynamic behavior of the structure. Therefore, in many cases the ANN is capable of

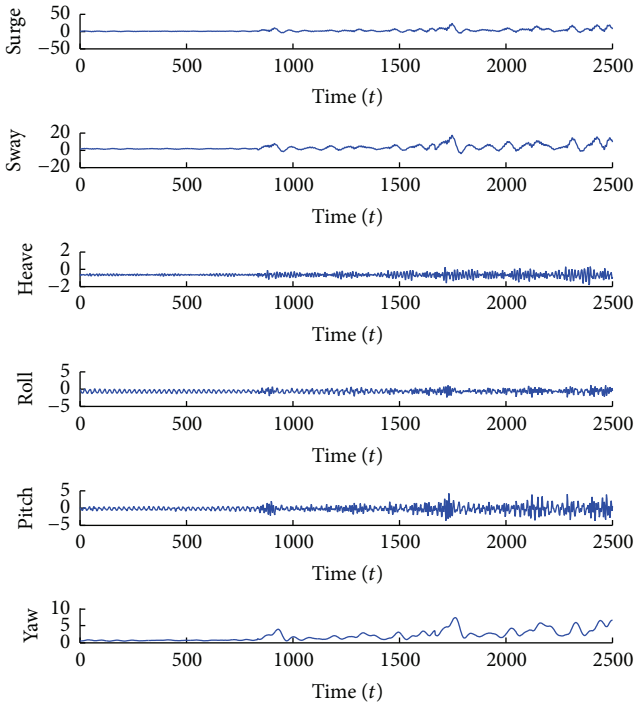


Figure 4: Platform motion used as ANN training input data.

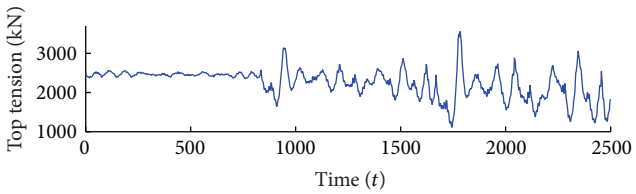


Figure 5: Tension force history used as ANN training target data.

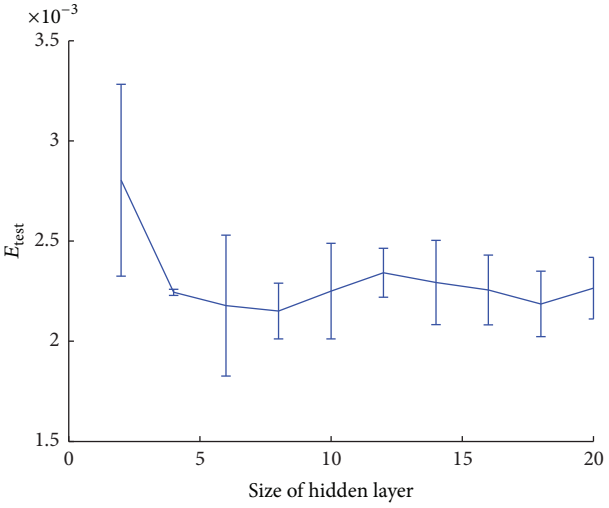


Figure 6: Validation error as function of number of neurons in the hidden layer.

handling fairly large time step increments compared to the corresponding numerical models. When using a larger time step in the ANN simulations, for example, by omitting a number of in-between data points, it is possible to reduce the size of the training data set and thereby reduce the computational time used for ANN training and eventually also for the ANN simulation. In the example of this paper a time step size of $h = 2.0$ s is found to yield a good balance between accuracy and computational efficiency, and this time step h is therefore used for the ANN simulations.

3.3. Design of ANN Architecture. In the design of the ANN architecture three variables are investigated: (1) number of neurons in the hidden layer, (2) size of the model memory d, and (3) required amount of training data. When the ANN has been trained and ready for use the network size has no significant influence on the total simulation time and computational effort. The main time consumer is the training part, and time used for training of the network highly depends on network size and the size of the training data set. Hence, it is of great interest to design an ANN architecture that is as compact and effective as possible.

Figure 6 shows a plot of the test error measure E_{test} relative to the number of neurons in the hidden layer of the ANN. In this section, where the three basic ANN variables are chosen the error measure is the mean square error (MSE), corresponding to the E_2 error measure in Section 2.3. The curve in Figure 6 furthermore represents the mean value of the error based on five simulations, while the vertical bars indicate the standard deviation. It is seen from this curve that the performance and the scatter in performance of the trained ANN is lowest when the hidden layer contains four neurons. Therefore, an effective and fairly accurate ANN performance is expected when four neurons in the hidden layer are used in the following simulations.

Figure 7 shows the test error relative to the model memory d, which represents the number of previous time steps used as network input. First of all it is found that including memory in the model significantly reduces the error. However, it is also seen from the figure that an increase of the memory beyond four previous steps implies no significant improvement in the ANN performance. Thus, a four-step memory, that is, $d = 4$, is used in the following numerical analyses.

For the training of any ANN it is always crucial to have a sufficient amount of training data in order to cover the full range of the network and secure applicability with sufficient statistical significance. Figure 8 shows the test error as function of the length of the training data set. As for the parameter studies in Figures 6 and 7 the present curve shows the mean results based on five simulation records. To make sure that a sufficient amount of data is used for the training of the ANN a total simulated record of 2500 s is included, which corresponds to approximately a length of 14 minutes for each of the three sea states. It is seen in Figure 8 that this length of the simulation record is more than sufficient to secure a low error.

The trained ANN is able to generate nonlinear output without equilibrium iterations and hence at an often significantly higher computational pace compared to classic integration procedures with embedded iteration schemes. Figure 9 shows the simulation of the top tension force in the mooring line calculated by the finite element method in RIFLEX and by the trained ANN. The four subfigures in Figure 9 represent the four wave heights that were not part of the ANN training, that is, $H_s = 4$ m, 6 m, 10 m, and 12 m. For these particular simulation records the trained ANN calculates a factor of about 600 times faster than the FEM calculations by RIFLEX.

3.4. Comparison of Error Measures. In the design of the ANN architecture presented above the results are obtained for an ANN trained with the MSE as objective function or error measure. It is in the following conveniently assumed that this ANN architecture is valid regardless of the specific choice of error function. Thus, the various error measures presented in Section 2.3 are in this section compared for the ANN with four neurons in the hidden layer, four memory input variables, and a training length of 2500 s.

As mentioned earlier some of the pregenerated data are saved for performance validation of the trained ANN. These data are used to calculate a validation error E_{test} which is the measure for the accuracy of the trained network. Figure 10 shows the development in the validation error during the network training with all four different error functions present in Section 2.3. It is clearly seen that all four error measures are minimized during training, whereas it is difficult to compare the detailed performance and efficiency of the four different networks based on these curves.

Figure 11(a) summarizes the development of the validation error for the four ANN, but this time the validation error is calculated using the same MSE error measure (E_2) to give a consistent basis for comparison. Thus, the four networks have been trained with four error measures, respectively, while in Figure 11(a) they are compared by the MSE. Even though the networks here are compared on common ground it is still difficult to evaluate how well they will perform individually on a full fatigue analysis. Figure 11(b) illustrates the accuracy of the four networks by showing a close up of a local maximum in the top tension force time history. It is seen that the two unweighted error functions, E_2 and E_4, perform superiorly compared to the weighted functions. Also the unweighted error measures provide a smaller MSE error in Figure 11(a). This indicates that weighting of the error functions implies no improvement of the performance and accuracy of the ANN.

3.5. Rain Flow Count. The magnitude of the various test error measures is difficult to relate directly to the performance of the ANN compared to the performance of the RIFLEX model. Since these long time-domain simulations are often used in fatigue life calculations an appropriate way to evaluate the accuracy of the ANN is to compare the accumulated rain

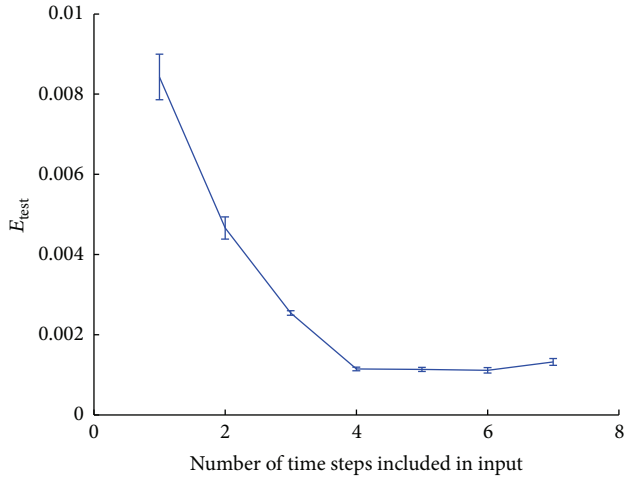


Figure 7: Validation error as function of model memory.

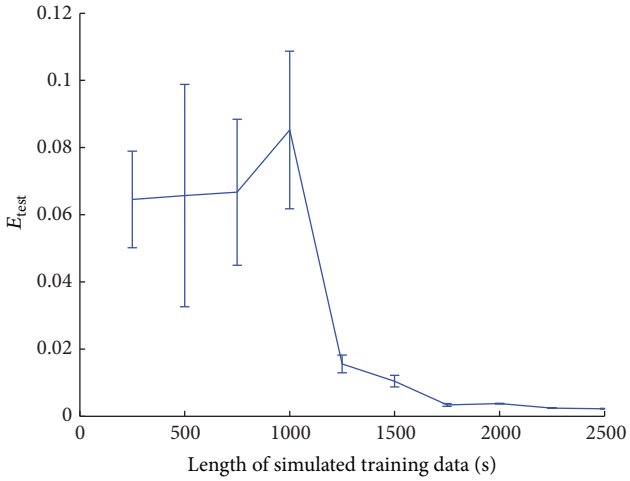


Figure 8: Validation error as function of amount of training data.

flow counts of the tension force cycles for each significant wave height. In these fatigue analyses the RIFLEX results are considered as the exact solution. For these calculations the full 3-hour simulations are used and Figure 12 shows the results of the rain flow count of accumulated tension force cycles for each of the significant wave heights. Deviations between RIFLEX and ANN simulations are listed in Table 1. It should be noted that the deviations for the individual seven sea states do not add up to give the total deviation because the individual sea states do not contribute equally to the overall damage. It is seen that the various networks perform very well on all individual sea states and that the best networks thereby obtain a deviation of less than 2% for the accumulated tension force cycles when summing up the contributions from all sea states. This deviation is a robust estimate and is likely to also represent the accuracy of a full subsequent fatigue life evaluation.

It is seen from the rain flow counting results in Figure 12 and Table 1 that the neural networks trained with unweighted error function in general perform slightly better than those trained with weighted error functions. Thus, placing a weight on the error function does not seem to have the desired effect in this application concerning the analysis of a mooring line system for a floating platform.

Table 1: Deviations on accumulated tension force cycles.

H_S	E_2	E_2^w	E_4	E_4^w
2	−3.6%	−16.3%	7.3%	65.8%
4	−6.9%	−13.0%	4.8%	32.0%
6	−4.4%	−8.0%	2.6%	14.9%
8	−2.7%	−4.8%	−1.5%	8.6%
10	−2.0%	−3.5%	−1.6%	6.9%
12	−1.4%	−3.1%	−1.7%	5.3%
14	−0.8%	−0.3%	2.1%	3.1%
Total	−1.7%	−3.8%	1.6%	6.4%

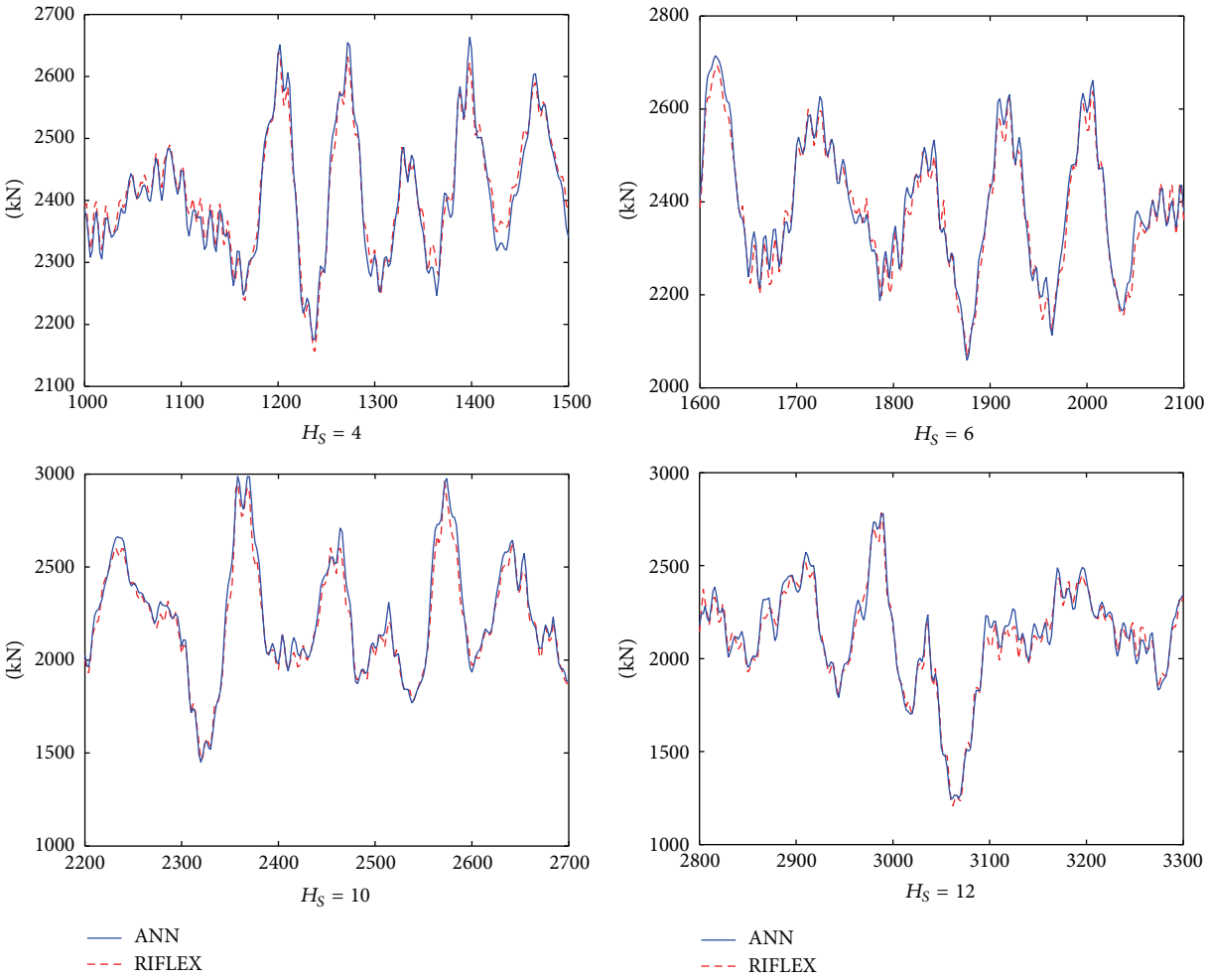


FIGURE 9: Comparison of simulated top tension forces using ANN and RIFLEX for the four wave heights that were not part of the network training.

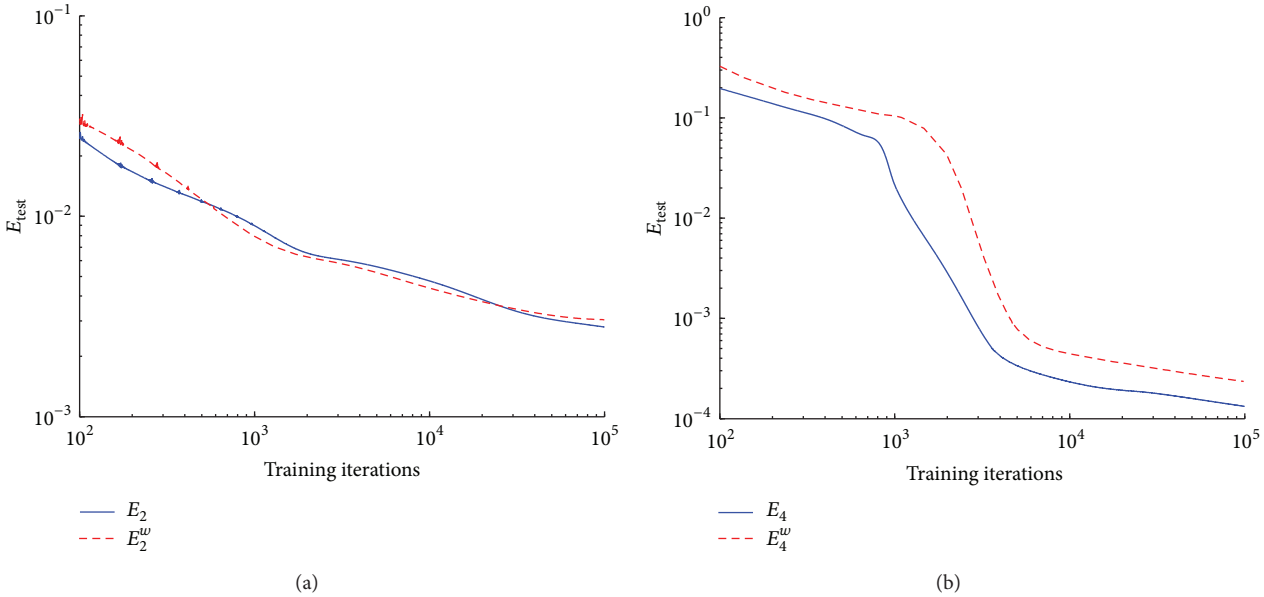
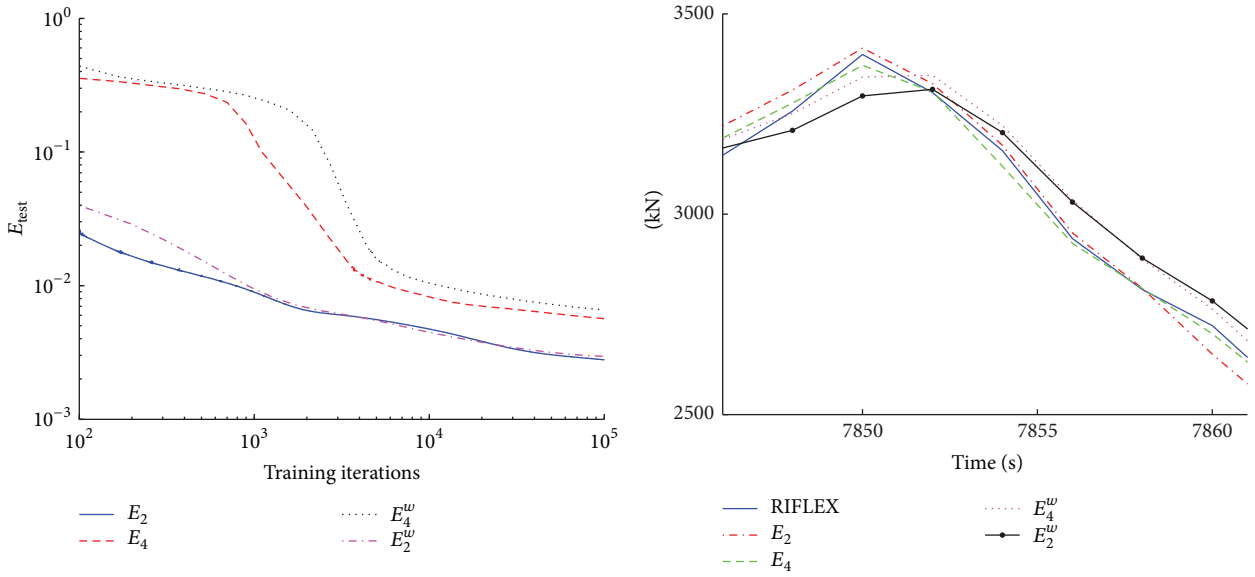


FIGURE 10: Validation error during training. (a) Error functions with $R = 2$; E_2, E_2^w. (b) Error functions with $R = 4$; E_4, E_4^w.



(a) Validation error during training

(b) Accuracy of simulation at peak

Figure 11: Comparison of performance for the four different error functions.

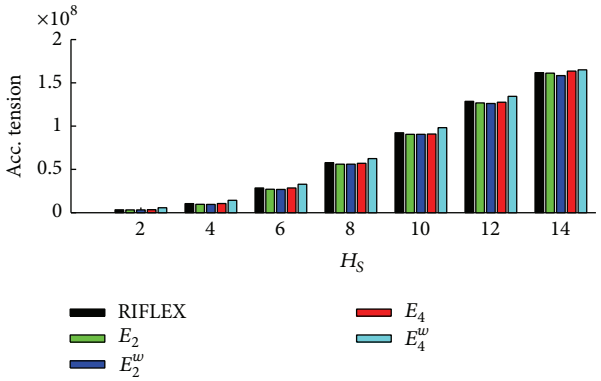


Figure 12: Accumulated stress for each sea state—calculated using rain flow counting.

4. Conclusion

It has been shown how a relatively small and compact artificial neural network can be trained to perform high speed dynamic simulation of tension forces in a mooring line on a floating platform. Furthermore, it has been shown that a proper selection of training data enables the ANN to cover a wide range of different sea states, even for sea states that are not included directly in the training data. In the example presented in this paper it is clear that weighting the error function used to train an ANN in order to emphasize peak response does not improve the network performance with respect to accuracy of fatigue calculations. In fact, the ANN appears to perform worse when trained with the weighted error function. On the other hand it appears that increasing the power of the error function from two to four provides a slight improvement to the performance of the trained ANN. However, the idea of a weighted error function seems to reduce the ANN performance. So apparently focusing on the high amplitudes seems to deteriorate the low amplitude response more than it improves the response with large amplitudes.

As a conclusion the Minkowski error with $R = 4$ is interesting for the mooring line example in more than one aspect. It provides more focus on the large amplitudes and improves the ANN slightly. Furthermore, the second derivative of the E_4 is fairly easy to determine, which makes this objective function suitable for several network optimizing schemes, such as Optimal Brain Damage (OBD) and Optimal Brain Surgeon (OBS), that are based on the second derivative of the error function. Network optimization is, however, not considered further in this paper but will be subject of future work.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] DNV, "Offshore Standard DNV-OS-F201—Dynamic Risers," Det Norske Veritas, October 2010.

[2] DNV, "Recommended Practice DNV-RP-F204—Riser Fatigue," Det Norske Veritas, July 2005.

[3] M. H. Patel and F. B. Seyed, "Review of flexible riser modelling and analysis techniques," *Engineering Structures*, vol. 17, no. 4, pp. 293–304, 1995.

[4] H. Adeli, "Neural networks in civil engineering: 1989–2000," *Computer-Aided Civil and Infrastructure Engineering*, vol. 16, no. 2, pp. 126–142, 2001.

[5] B. Warner and M. Misra, "Understanding neural networks as statistical tools," *The American Statistician*, vol. 50, no. 4, pp. 284–293, 1996.

[6] K. Ordaz-Hernandez, X. Fischer, and F. Bennis, "Model reduction technique for mechanical behaviour modelling: efficiency criteria and validity domain assessment," *Journal of Mechanical Engineering Science*, vol. 222, no. 3, pp. 493–505, 2008.

[7] M. Hosseini and H. Abbas, "Neural network approach for prediction of deflection of clamped beams struck by a mass," *Thin-Walled Structures*, vol. 60, pp. 222–228, 2012.

[8] R. Guarize, N. A. F. Matos, L. V. S. Sagrilo, and E. C. P. Lima, "Neural networks in the dynamic response analysis of slender marine structures," *Applied Ocean Research*, vol. 29, no. 4, pp. 191–198, 2007.

[9] N. H. Christiansen, P. E. T. Voie, J. Høgsberg, and N. Sødahl, "Efficient mooring line fatigue analysis using a hybrid method time domain simulation scheme," in *Proceedings of the 32nd ASME International Conference on Ocean, Offshore and Arctic Engineering (OMAE '13)*, vol. 1, 2013.

[10] S. A. Solla, E. Levin, and M. Fleisher, "Accelerated learning in layered neural networks," *Complex Systems*, vol. 2, no. 6, pp. 625–639, 1988.

[11] J. B. Hampshire and A. H. Waibel, "Novel objective function for improved phoneme recognition using time-delay neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 2, pp. 216–228, 1990.

[12] H. Altincay and M. Demirekler, "Comparison of different objective functions for optimal linear combination of classifiers for speaker identification," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '01)*, vol. 1, pp. 401–404, May 2001.

[13] MARINTEK, *Simo Theory Manual*, Sintef, Trondheim, Norway, 2009.

[14] MARINTE, *Riflex Theory Manual*, Sintef, Trondheim, Norway, 2008.

[15] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, NY, USA, 2006.

[16] C. M. Bishop, *Neural Networks for Pattern Recognition*, The Clarendon Press, Oxford, UK, 1995.

[17] R. Fletcher, *Practical Methods of Optimization*, John Wiley & Sons, Chichester, UK, 2nd edition, 1987.

Paper[P3]

Optimization of Neural Networks for Simulation of Slender Marine Structures

N. H. Christiansen, P. E. T. Voie, O. Winther, J. Høgsberg

Institution of Mechanical Engineers. Proceedings. Part M: Journal of Engineering for the Maritime Environment, (2015)

Review Article

Institution of MECHANICAL ENGINEERS

Journal of ENGINEERING FOR THE MARITIME ENVIRONMENT

Proc IMechE Part M:
J Engineering for the Maritime Environment
1–10

DOI: 10.1177/1475090215573090
pim.sagepub.com
SAGE

Optimization of neural networks for time-domain simulation of mooring lines

Niels H Christiansen[1,2], Per Erlend Torbergsen Voie[3], Ole Winther[4] and Jan Høgsberg[2]

Abstract

When using artificial neural networks in methods for dynamic analysis of slender structures, the computational effort associated with time-domain response simulation may be reduced drastically compared to classic solution strategies. This article demonstrates that the network structure of an artificial neural network, which has been trained to simulate forces in a mooring line of a floating offshore platform, can be optimized and reduced by different optimization procedures. The procedures both detect and prune the least salient network weights successively, and besides trimming the network, they also can be used to rank the importance of the various network inputs. The dynamic response of slender marine structures often depends on several external load components, and by applying the optimization procedures to a trained artificial neural network, it is possible to classify the external force components with respect to importance and subsequently determine which of them may be ignored in the analysis. The performance of the optimization procedures is illustrated by a numerical example, which shows that, in particular, the most simple procedures are able to remove more than half of the network weights in an artificial neural network without significant loss of simulation accuracy.

Keywords

Nonlinear dynamics, slender marine structures, neural networks, optimal brain damage, optimal brain surgeon

Date received: 20 August 2014; accepted: 15 January 2015

Introduction

In recent years, an increasing amount of machine learning techniques has proven to be useful in many aspects of structural engineering. Especially, the use of artificial neural networks (ANNs) has drawn vast attention over the last decades. A proper introduction and overview of a variety of neural network applications in civil engineering is given by Adeli.[1] In most situations, machine learning techniques are used to extract information from large datasets, which means that they are used for pattern identification. The ability to recognize patterns can be very useful in, for example, damage detection on existing structures where the ANN can be used to detect changes in the structural behavior due to possible damage in structural members, see Bakhary et al.[2] for further details.

Furthermore, neural networks hold a great potential in structural design analysis because they are capable of performing nonlinear dynamic analysis at a competitive pace. An example is provided by Guarize et al.[3] who showed how a hybrid method, combining the finite element method (FEM) and an ANN, can simulate the dynamic response of a flexible oil pipe in service about 20 times faster than an ordinary nonlinear numerical analysis. A similar approach has recently been used by Christiansen et al.[4] to reduce the simulation time dramatically when conducting a full fatigue life analysis of a mooring line on a floating platform. The same platform structure has been used in Christiansen et al.[5] to investigate the influence of using different error measures or cost functions in the network training, and it was, furthermore, found that the resulting trained

[1]DNV GL, Hellerup, Denmark
[2]Department of Mechanical Engineering, Technical University of Denmark, Lyngby, Denmark
[3]DNV GL, Trondheim, Norway
[4]Department of Applied Mathematics and Computer Science, Technical University of Denmark, Lyngby, Denmark

Corresponding author:
Niels H Christiansen, DNV GL, Tuborg Parkvej 8, Hellerup 2900, Denmark.
Email: niels.christiansen@dnvgl.com

network used in that analysis simulated more than two orders of magnitude faster than a corresponding nonlinear numerical analysis.

The aim of this study is to show that the ANN presented in Christiansen et al.[5] can, in fact, be further optimized and reduced in size. The literature suggests a broad variety of pruning methods for optimizing the ANN architecture, and an overview over various pruning algorithms is given in the survey article of Reed.[6] This study considers the simplest possible pruning strategy, which assumes that small weights are less important than large weights, and two of the most prominent procedures: optimal brain damage (OBD)[7] and optimal brain surgeon (OBS).[8] The procedures effectively identify and prune the least salient network weights in an ANN. The OBD procedure was introduced by Le Cun et al.[7] in the early 1990s, where it was used to optimize an ANN that was trained to read handwritten and printed zip code digits. The OBS procedure was presented by Hassibi et al.[8] some years later. The OBS follows the same line of philosophy as the OBD but uses a different approach that eliminates some of the limiting assumptions of the original OBD procedure.

The three procedures are tested in this article on data generated by a numerical (FEM) analysis of the mooring lines on a floating offshore platform structure. Based on dynamic time-domain simulations of the platform, the ANN is trained to predict the relationship between platform motion and tension forces in the mooring lines. The performance of the ANN is evaluated with respect to common error measures and fatigue damage for the most critical mooring lines. It is demonstrated that the performance of the ANN can be improved by deleting the least salient network connections identified by the various pruning procedures. It is demonstrated that when using the simple procedure or the OBD procedure, it is possible to eliminate more than half of the network connections in the ANN without impairing the simulation accuracy significantly.

ANN

The ANN is a machine learning tool with a simple structure inspired by real neural circuits. This structure makes the ANN capable of recognizing patterns in large datasets. This ability can be utilized to perform nonlinear simulations, often at a very high computational rate. A thorough introduction to neural networks, and their various features, is given by Warner and Misra.[9] In the following, an ANN which has been trained to recognize and predict the relationship between the motion of an offshore floating platform and the resulting tension forces in a selected mooring line will be optimized by the use of the three pruning procedures mentioned previously. For additional details concerning the ANN used in this article, see Christiansen et al.[5]

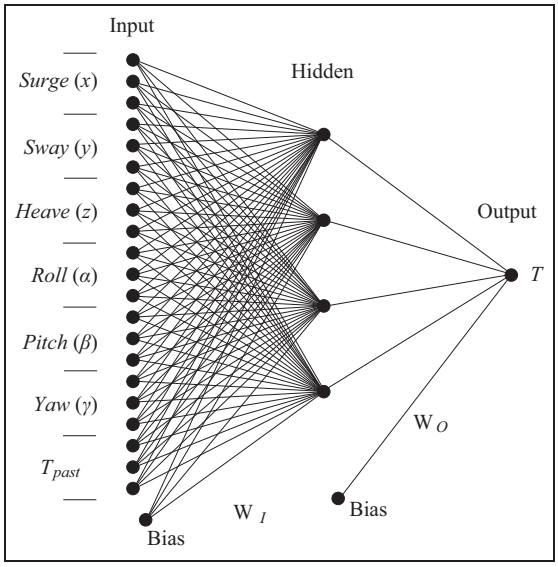


Figure 1. Sketch of artificial neural network for predicting top tension force in a mooring line.

Architecture of ANN

The architecture of a typical single-layer ANN is shown in Figure 1. The ANN consists of an input layer, where each input neuron represents a measured time discrete state of the system. In the present case, in Figure 1, the neurons of the input layer represent the six motion components (x, y,…) of the floating offshore platform (as illustrated in Figure 3) and the previous time discrete values of the mooring line tension force (T_{past}). The input layer is connected to a single hidden layer, which is then connected to the output layer representing the current time value of the tension force (T). In the network, neurons in neighboring layers are connected, and each of these connections carries a weight. The training of an ANN corresponds to an optimization of these weights with respect to a particular data training set. The accuracy and efficiency of the network depend on the network architecture, the optimization of the individual weights and the choice of error function used in the applied optimization procedure.

The design and architecture of the ANN and the subsequent training procedure both follow the approach outlined in Bishop.[10] Assume that the vectors $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ contain the neuron variables of the input layer, output layer and hidden layer, respectively. The output layer and hidden layer values can be calculated by the expressions

$$\mathbf{y} = \mathbf{W}_O^{\mathrm{T}}\mathbf{z}, \quad \mathbf{z} = \tanh\left(\mathbf{W}_I^{\mathrm{T}}\mathbf{x}\right), \quad x_0 \equiv z_0 \equiv 1 \tag{1}$$

where $\mathbf{W}_I$ and $\mathbf{W}_O$ are arrays that contain the neuron connection weights between the input and the hidden layers and the hidden and the output layers, respectively. By having both x_0 and z_0 defined equal to unity biases in the data can be absorbed by the input and hidden layers. The tangent hyperbolic function is used as

an activation function between the input and the hidden layers. A nonlinear activation function is needed in order to introduce nonlinearities into the neural network. The tangent hyperbolic function is often used in networks performing regression between continuous variables because it provides fast convergence in the network training procedure.[11]

The optimal weight components of $\mathbf{W}_I$ and $\mathbf{W}_O$ are found by an iterative procedure, where the weights are modified to give a minimum with respect to a certain error function. In the present case, the Minkowski-R error[12] with $R = 4$ is used. In Christiansen et al.,[5] it has been demonstrated that this particular error function yields good ANN performance for the offshore application considered in this article. The Minkowski-4 error function can be written as

$$E(\mathbf{W}) = \frac{1}{4}\sum_{n=1}^{N}\{y(\mathbf{x}_n;\mathbf{W}_I,\mathbf{W}_O) - t_n\}^4 \tag{2}$$

where N is the number of training datasets, y is the ANN output obtained by combining the individual contributions in equation (1), while t is the associated target value. In equation (2), the vector $\mathbf{W}$ contains the weight components of $\mathbf{W}_I$ and $\mathbf{W}_O$, respectively.

The updating of the weight components is performed by a classic gradient decent technique, which adjusts the weights in the opposite direction of the gradient of the error function, see, for example, Fletcher[13] for details on conjugate gradient optimization. For the ANN, this gradient decent updating can be written as

$$\mathbf{W}_{new} = \mathbf{W}_{old} + \Delta\mathbf{W}, \quad \Delta\mathbf{W} = -\eta\frac{\partial E(\mathbf{W})}{\partial\mathbf{W}} \tag{3}$$

where η is the learning step size parameter. This parameter can either be constant or updated during the training of the ANN. For the applications in this article, the learning step size parameter is dynamic and will be adjusted for each iteration, so that it increases when the training error decreases compared to previous iteration step, while it decreases when the error increases. In the weight update (3), the increment $\Delta\mathbf{W}$ must be evaluated separately for the components of $\mathbf{W}_I$ and $\mathbf{W}_O$.

Network optimization algorithms

The aim of the optimization procedure in this article is initially to classify the layer connections in terms of relevance and subsequently to use this classification to remove connections and thereby improve the efficiency of the ANN. Thus, the optimization procedures initiate from a reasonably large network that has been trained to minimum error. Although the ANN used in the present case already is rather compact in terms of size of input and hidden layer, it is assumed that the network is still sufficiently large to be pruned further on individual weight level.

There exist several methodologies for evaluating the saliency of the network weights.[11] The idea of the pruning methods is to rank the importance of all network weights and then successively delete the least salient weight. The simplest possible approach is to simply consider the magnitude of all network weights and assume that small weights are less important than large weights. This approach requires that all input and output data are normalized prior to the training of the ANN, so that all datasets have the same range of magnitude. Thereby, the ranking of network weights becomes embedded in the training as it is assumed that the ANN training procedure "ignores" the least important inputs by reducing the magnitude of the associated weights. The normalization of data is done regardless of the optimization procedures, as it has shown to lead to better and faster convergence. However, according to Bishop,[11] this concept has little theoretical motivation. Nevertheless, for comparison, this method is implemented and tested in the following together with the two previously mentioned algorithms OBD and OBS.

The starting point for both OBD and OBS is a Taylor expansion of the error function given in equation (2) with respect to the network weights. This Taylor series can be written as

$$\delta E = \frac{\partial E}{\partial\mathbf{W}^T}\delta\mathbf{W} + \frac{1}{2}\delta\mathbf{W}^T\mathbf{H}\delta\mathbf{W} + O\left(||\delta\mathbf{W}||^3\right) \tag{4}$$

where

$$\mathbf{H} = \frac{\partial^2 E}{\partial\mathbf{W}\partial\mathbf{W}^T} \tag{5}$$

is the Hessian matrix containing all second-order derivatives of the error function with respect to the network weights. Both optimization methods are based on networks that are initially trained to a local minimum. This means that the first term in equation (4) vanishes because $\partial E/\partial\mathbf{W}^T = 0$ represents the condition for a function extremum. The higher-order terms represented by the latter term in equation (4) are furthermore omitted. This implies that the saliency of the network weights is calculated merely by evaluating the second-order derivatives, whereby the problem reduces to

$$\delta E \simeq \frac{1}{2}\delta\mathbf{W}^T\mathbf{H}\delta\mathbf{W} \tag{6}$$

This approximation of the perturbation of the error function gives the curvature in error space at the local minimum that was reached during the network training. The particular diagonal element in the Hessian matrix with smallest magnitude identifies the weight component with least significance with respect to the error function. This weight is, therefore, deleted by the pruning procedure. Figure 2 shows a two-dimensional example, where the weight $W_{i,j}$ is less significant compared to $W_{k,l}$ because $(1/2)(\partial^2 E/\partial W_{i,j}^2)W_{i,j}^2 < (1/2)(\partial^2 E/\partial W_{k,l}^2)W_{k,l}^2$. The expression in equation (6) constitutes the foundation for both the OBD and the OBS procedures, even though the two methods use

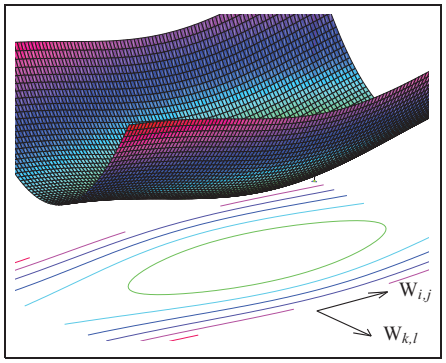


Figure 2. Error function around a local minimum. The curvature of the error surface in each direction corresponds to the saliency of the associated weight. In this illustration, we see that $(1/2)(\partial^2 E/\partial W_{i,j}^2)W_{i,j}^2 < (1/2)(\partial^2 E/\partial W_{k,l}^2)W_{k,l}^2$.

different approaches to determine the Hessian matrix $\mathbf{H}$, as demonstrated in the following.

OBD

The OBD procedure trains the initial ANN architecture, where all neurons between input and hidden layers and between hidden and output layers are connected with nonvanishing weights. The saliency of all weights is computed, and the least important weight is pruned by simply defining the associated weight to be 0. When a network weight is eliminated, the reduced network must be retrained before the next least salient weight can be selected and subsequently removed. The procedure can, in principle, be repeated until the network is completely eliminated, whereas in reality, the procedure is terminated when a desired compromise between accuracy and computational efficiency is obtained.

The OBD procedure approximates the Hessian by disregarding the nondiagonal terms. With the error function given in equation (2), this assumption yields the following second derivatives of the network error with respect to the components of the weight matrices $\mathbf{W}_O$ and $\mathbf{W}_I$, respectively. The diagonal Hessian components for the hidden-to-output layer connections can be computed as

$$\frac{\partial^2 E}{\partial W_{O,j}^2} = 3\sum_{n=1}^{N}(y_n - t_n)^2 z_j^2 \tag{7}$$

while for the input-to-hidden layer connections

$$\frac{\partial^2 E}{\partial W_{I,ji}^2} = \sum_{n=1}^{N}\left(1 - z_j^2\right)\left(3\left(1 - z_j^2\right)(y_n - t_n)^2 W_{0,ij} - 2z_j(y_n - t_n)^3\right)W_{0,ji}x_i^2 \tag{8}$$

Having these double derivatives determined by equations (7) and (8), it is possible to evaluate the saliency for all individual network weights as

$$L_q^{OBD} = \frac{1}{2}\mathbf{H}_{qq}W_q^2 \tag{9}$$

where the subscript represents the qth weight. Thus, the least salient weight can be identified by L_q^{OBD} in equation (9) and subsequently deleted.

OBS

The goal of the OBS method is to minimize the increase in the error function given in equation (2) while setting a specific network weight to 0. The vanishing weight, which represents the candidate for deletion, is in the following denoted as W_q. The condition for elimination of the weight with the smallest saliency can be expressed by the projection

$$\mathbf{e}_q^T\delta\mathbf{W} + W_q = 0 \tag{10}$$

where $\mathbf{e}_q$ is a zero vector in weight space with unit value corresponding to weight W_q. The aim is now to solve the following equation, which minimizes the increase in error while forcing the selected weight to 0

$$\min_q\left\{\min_{\delta\mathbf{W}}\left(\frac{1}{2}\delta\mathbf{W}^T\mathbf{H}\delta\mathbf{W}\right)|\mathbf{e}_q^T\delta\mathbf{W} + W_q = 0\right\} \tag{11}$$

To solve equation (11), a Lagrangian is formed containing $\delta E = 0$, with the perturbation of the error function given in equation (6), and the condition in equation (10) enforced by the Lagrange multiplier λ

$$L = \frac{1}{2}\delta\mathbf{W}^T\mathbf{H}\delta\mathbf{W} + \lambda\left(\mathbf{e}_q^T\delta\mathbf{W} + W_q\right) \tag{12}$$

When deriving the functional derivatives of this Lagrangian, and employing the constraint condition in equation (10), the perturbation of the weight matrix is obtained as

$$\delta\mathbf{W} = -2\lambda\mathbf{H}^{-1}\mathbf{e}_q \tag{13}$$

This is now inserted into equation (10), which then gives the Lagrange multiplier λ as

$$\lambda = \frac{W_q}{2\left[\mathbf{H}^{-1}\right]_{qq}} \tag{14}$$

where $[\mathbf{H}^{-1}]_{qq} = \mathbf{e}_q^T\mathbf{H}^{-1}\mathbf{e}_q$ represents the diagonal element of the inverse Hessian corresponding to the weight component q. To calculate the optimal change in weight, the value of λ is finally inserted into equation (13), which gives

$$\delta\mathbf{W} = -\frac{W_q}{\left[\mathbf{H}^{-1}\right]_{qq}}\mathbf{H}^{-1}\mathbf{e}_q \tag{15}$$

Now that this optimal change in weight has been determined, and the resulting change in the error is obtained by inserting equations (14) and (15) into equation (12)

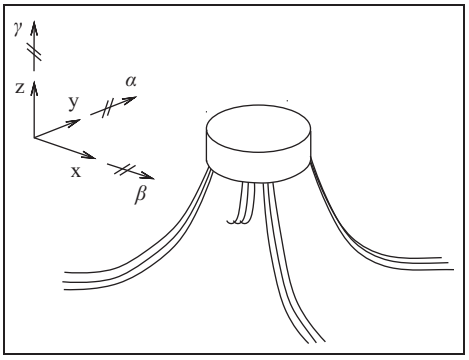


Figure 3. Mooring system with floating platform and anchor lines.

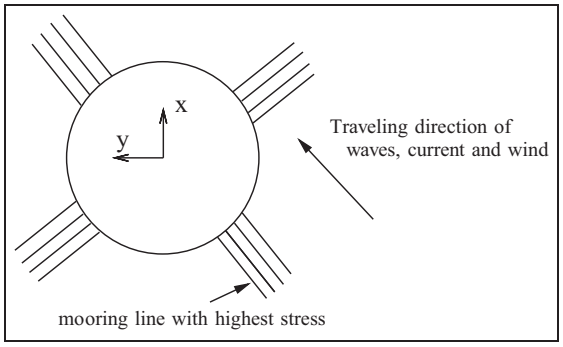


Figure 4. Mooring line system (top view).

$$L_q^{OBS} = \frac{1}{2}\frac{W_q^2}{\left[\mathbf{H}^{-1}\right]_{qq}} \tag{16}$$

As seen, the inverse Hessian is needed for calculating both the smallest saliency L_q and the corresponding optimal change in weight $\delta\mathbf{W}$. Hassibi et al.[8] outline a procedure for the direct evaluation of the inverse Hessian matrix $\mathbf{H}^{-1}$, which avoids the computational effort associated with the inversion of large matrices. However, since the weight matrices used in the example for this article are fairly small, the inversion of $\mathbf{H}$ is based on standard numerical techniques without special considerations.

Marine structural model

The structure used for ANN training and simulation is sketched in Figure 3. It consists of a floating offshore platform located at 105-m water depth and anchored by 18 mooring lines, which are assembled in four main clusters. The external forces acting on the structure are induced by waves, current and wind.

In principle, the dynamic analysis of the platform–mooring system corresponds to time integration of the discretized equation of motion

$$\mathbf{M}(\mathbf{r})\ddot{\mathbf{r}}(t) + \mathbf{C}(\mathbf{r})\dot{\mathbf{r}}(t) + \mathbf{K}(\mathbf{r})\mathbf{r}(t) = \mathbf{f}(t) \tag{17}$$

In this nonlinear equation, $\mathbf{r}$ contains the degrees of freedom of the structural model, $\mathbf{f}$ includes all external forces acting on the structure from, for example, gravity, buoyancy and hydrodynamic effects, while the nonconstant matrices $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{K}$ represent the system inertia, damping and stiffness, respectively. The system inertia matrix $\mathbf{M}$ takes into account both the structural inertia and the response-dependent hydrodynamic added mass. Linear and nonlinear energy dissipations from both internal structural damping and hydrodynamic damping are accounted for by the state-dependent damping matrix $\mathbf{C}$. Finally, the stiffness matrix contains contributions from both the constitutive elastic stiffness and the response-dependent geometric stiffness.

The nonlinear equations of motion in equation (17) couple the structural response of the floating platform and the response of the mooring lines. However, the system is effectively solved by separating the solution procedure into the following steps. First, the motion of the floating platform is computed by the program SIMO,[14] assuming a quasi-static catenary mooring line model with geometric nonlinearities. The platform response from this initial analysis is subsequently used as excitation in terms of prescribed platform motion in a detailed nonlinear finite element analysis for the specific mooring line with largest tension stresses. The location of the mooring line with largest stresses is indicated in Figure 4. For this specific line, the hot spot with respect to fatigue is located close to the platform, and it is in the following referred to as the top tension force. From the detailed fully nonlinear analysis performed by RIFLEX,[15] the time history of the top tension force at this hot spot is extracted. Seven different 3-h time-domain simulations are generated. The simulations cover seven different sea states with 2-, 4-,..., 14-m significant wave height, respectively. All sea states have a peak period of 10 s. In the following, these time histories are used for training and validation of the ANN and for demonstration of the various optimization techniques.

Results

In Christiansen et al.,[5] an ANN was trained to predict the top tension forces in the selected mooring line with the platform response variables as network input. The network training was based on the simulated time histories for both the platform motion and the top tension force. In Christiansen et al.,[4] a multilayer ANN was trained to simulate the top tension force two orders of magnitude faster than a corresponding time integration solution of the numerical model. The training data were set up and arranged, so that a simple ANN with a single hidden layer could simulate all fatigue relevant sea states and thereby provide a significant reduction in the computational effort associated with a fatigue life evaluation. For clarity, the ANN used in Christiansen et al.[5] covered only a few sea states, with different

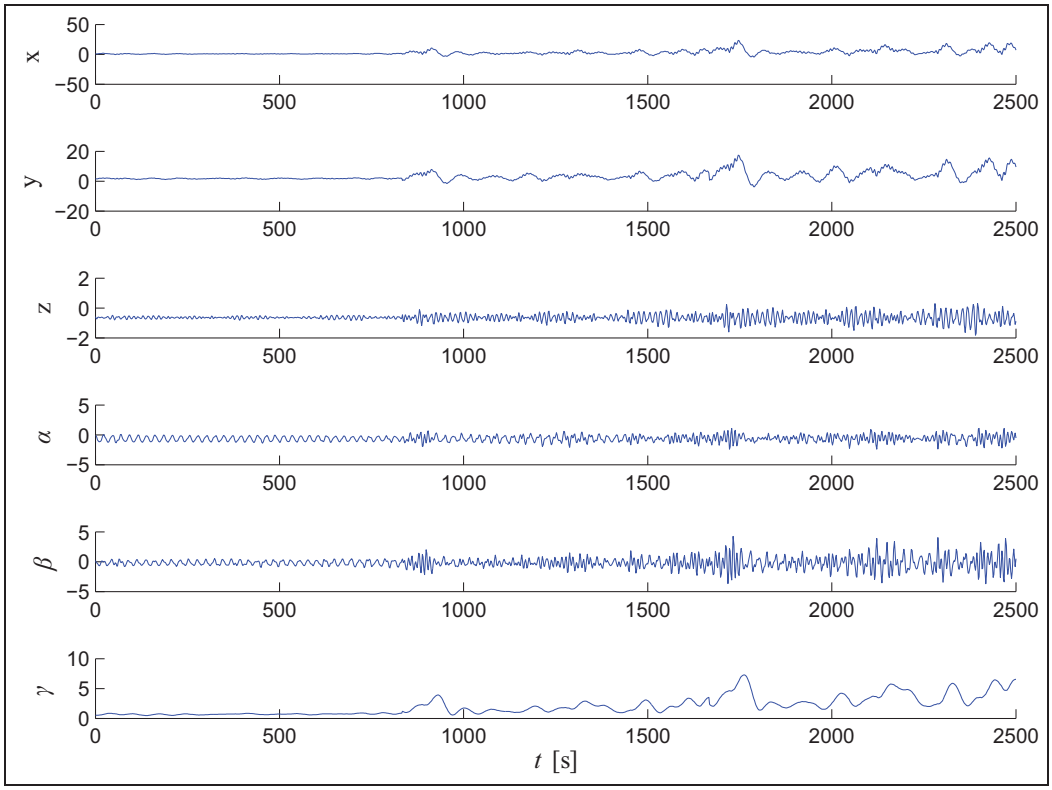


Figure 5. Platform motion used as ANN training input data. Dimensions are in meters for *x*, *y* and *z* and in degrees for α, β and γ.

significant wave heights and constant peak period. This gave a compact neural network, conveniently reused in this article to illustrate the influence of applying the various optimization algorithms.

Network design

Optimal network architecture and design for this application have been investigated in Christiansen et al.,[5] where it is demonstrated that a network with four neurons in the hidden layer and a model memory of three time steps give a compact and yet very accurate ANN with a total of 93 network weights, as shown in Figure 1. Furthermore, it was found that it is beneficial to use a different time step size than the one used in the RIFLEX simulations. In order to keep the associated Newton–Raphson iteration algorithm stable, RIFLEX uses a time step size of 0.1 s. However, it was shown in Christiansen et al.[5] that a time step size of $h = 2$ is sufficient for the ANN to grasp the dynamic behavior of the structure. When using the platform motion components together with the top tension force from previous time steps as the ANN input, the input network vector can be written as

$$\begin{aligned}\mathbf{x}_n = [[&[x_t \quad x_{t-h} \quad x_{t-2h}]\ [y_t \quad y_{t-h} \quad y_{2t-h}]\\ &[z_t \quad z_{t-h} \quad z_{2t-h}]\ [\alpha_t \quad \alpha_{t-h} \quad \alpha_{t-2h}]\\ &[\beta_t \quad \beta_{t-h} \quad \beta_{t-2h}]\ [\gamma_t \quad \gamma_{t-h} \quad \gamma_{t-2h}]\\ &[T_{t-h} \quad T_{t-2h} \quad T_{t-3h}]]^T\end{aligned} \tag{18}$$

where T denotes the top tension force; t is the current time; h is the time increment; and x, y, z, α, β and γ represent the platform motion components shown in Figure 3: surge, sway, heave, roll, pitch and yaw, respectively. The corresponding ANN output is the current top tension force T_t

$$y_n = T_t \tag{19}$$

Since there is only a single network output in this analysis, y is now a scalar quantity and not a vector as originally assumed in equation (1). For the training of the ANN, nonlinear simulations in RIFLEX are conducted for sea states with a significant wave height of $H_s = 2$, 8 and 14 m, respectively. With these wave heights included in the training data, the ANN is able to provide accurate time histories for the top tension force T for all intermediate wave heights as well.[5] The data used for training of the ANN are shown in Figures 5 and 6. Figure 5 shows the time histories for the six motion degrees of freedom of the platform calculated by the initial analysis in SIMO and subsequently used as input to both the full numerical analysis in RIFLEX and the ANN training and simulation. Figure 6 shows the corresponding time history of the top tension force T determined by RIFLEX. The full time histories shown in Figures 5 and 6 are constructed of time series for the three significant wave heights. It should be mentioned that the initial 830 s of the training dataset represents a significant wave height $H_s = 2$ m, the intermediate 830-s interval is for

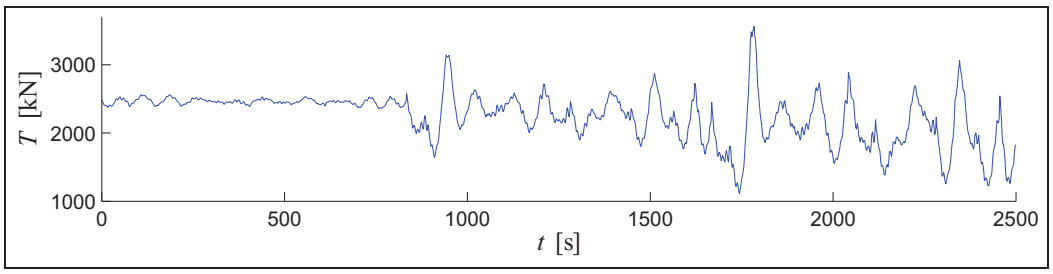


Figure 6. Tension force history used as ANN training target data.

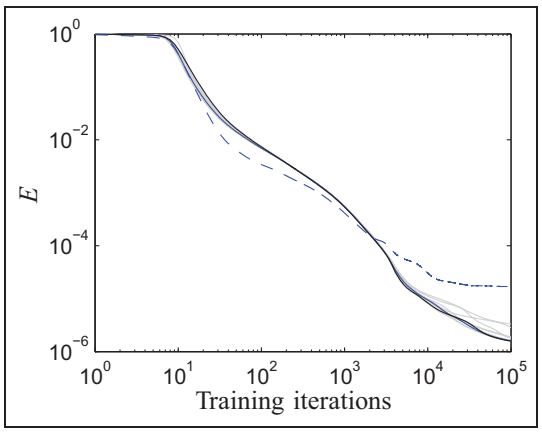


Figure 7. Development in ANN error E during training.

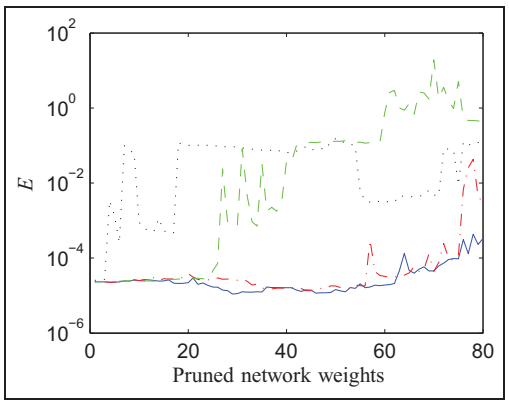


Figure 8. Development in error E as number of pruned network weights increases: simple, OBD, OBS and random.

$H_s = 8$ m, while the remaining final part of the dataset is for $H_s = 14$ m. Before being fed into the network, all data are normalized, so that mean values are equal to 0 and all standard deviations are equal to 1. This leads to better and faster convergence during the network training.

Optimization of network

As mentioned previously, it is one of the basic assumptions for the optimization methods that the pruning procedures initiate from a network which has been trained to a local minimum. Hence, the first step when applying the pruning methods is to train the ANN. Figure 7 shows the development in ANN error E from equation (2) during training. It is assumed that 10^5 training iterations are sufficient to assume that the network has reached a minimum in the network error E. The solid lines in the graph are the results of 10 runs. The bold line represents the ANN which is used in the subsequent optimization. The dashed line shows the corresponding test error E_{test} calculated on the test data which are not part of the training dataset. The test error indicates the starting point for the optimization in Figure 8. The three optimization algorithms have been applied to the trained ANN.

Figure 8 shows the development of the ANN error as the number of deleted network weights increases. To verify that the three procedures actually select and delete appropriate weights, a fourth algorithm, which randomly deletes weights, is also included in Figure 8 (dotted line). For the simple approach, OBD and random deletion of weights, the network is retrained each time a weight is deleted, whereas for the OBS, the network is trained only once, whereupon the network weights are adjusted according to the procedure described in section "OBS," and therefore, this procedure requires no additional network training. Apparently, this procedure for updating weights without additional training holds only for a limited number of deleted weights in this particular example. The initial network training requires a fairly large amount of iterations and takes about 10 min before a local minimum in error is reached. However, the retraining of the network performed after each weight deletion does not have to be as thorough. In this example, 2000 iterations for the retraining have proven to give satisfactory results, which means that the retraining is not very time-consuming. Hence, the retraining may, therefore, be considered as a small adjustment to the weights and not a complete network training. This means that the OBD also takes in the order of 10 min. The OBS procedure does not involve any retraining. However, as the evaluations of the inverse Hessian $\mathbf{H}^{-1}$ is done via a sequential summation over all training data for each step in the pruning procedure,[8] the computation time saved compared to OBD is insignificant. Even though retraining should be unnecessary in the OBS procedure, it is recommended that the final network is retrained

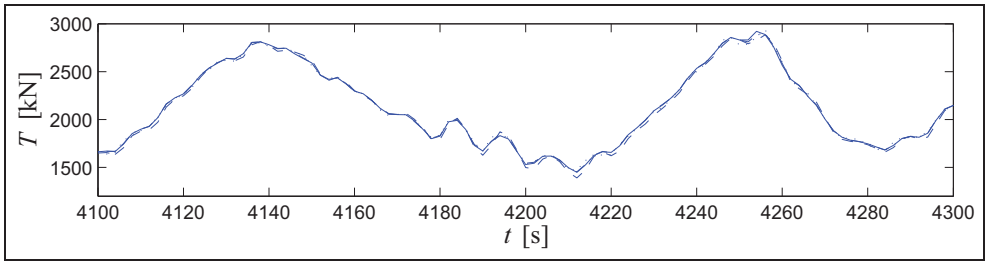


Figure 9. Simulation after 20 pruned connections: ·–·–·– simple, ——— OBD, – – – OBS and ..... RIFLEX.

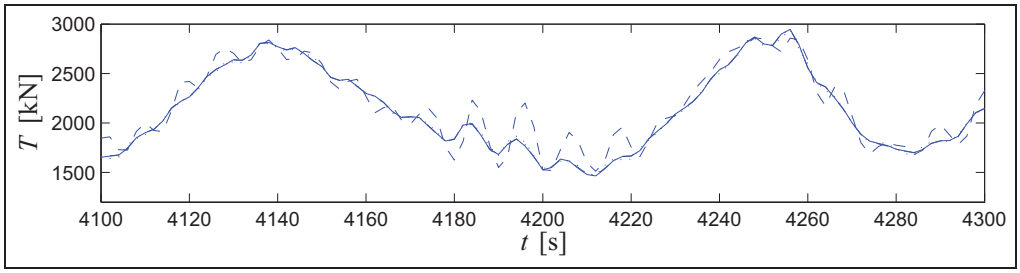


Figure 10. Simulation after 40 pruned connections: ·–·–·– simple, ——— OBD, – – – OBS and ..... RIFLEX.

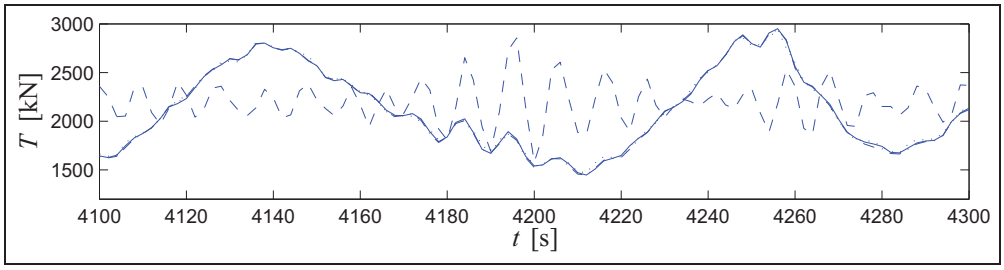


Figure 11. Simulation after 60 pruned connections: ·–·–·– simple, ——— OBD, – – – OBS and ..... RIFLEX.

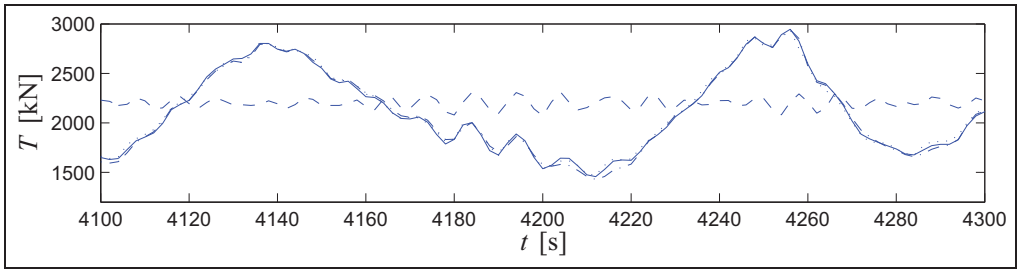


Figure 12. Simulation after 70 pruned connections: ·–·–·– simple, ——— OBD, – – – OBS and ..... RIFLEX.

after the pruning exercise has been terminated. The performance of the networks pruned by the OBS procedure is shown in Figure 8 by the green dashed curve. It is seen that the error E of the OBS procedure is relatively low for up to approximately 25 pruned weights. Although simulations presented by Hassibi et al.[8] indicate that the OBS procedure is superior to OBD, it appears from the present analysis that the OBD (solid blue line) procedure is apparently the best pruning procedure in this particular example concerning dynamic analysis of a slender offshore structure. On the other hand, the very simple approach (dashed-dotted red curve), where weights are directly deleted according to their magnitude, seems to work very well on this particular example despite its alleged poor performance in practice.[11]

It is not straightforward to relate an error measure to the accuracy of the corresponding ANN simulation. Figures 9–12 compare the time integration solutions obtained by RIFLEX to the simulations performed by the ANNs optimized by the three pruning algorithms. These figures show the top tension force for the same time interval at different stages of the pruning

Table 1. Deviations on accumulated tension force cycles when compared to RIFLEX simulations.

Pruned weights	0	20	40	60	70	Optimal[a]
Simple	1.4%	1.0%	−1.0%	−1.5%	−4.7%	0.04%
OBD	1.4%	1.3%	−0.3%	−2.3%	−0.4%	0.06%
OBS	1.4%	7.5%	49.0%	60.6%	377.0%	1.0%

OBD: optimal brain damage; OBS: optimal brain surgeon.
[a]Optimal ANN in test during pruning: simple procedure after 45 pruned weights, OBD after 29 pruned weights and OBS after 4 pruned weights.

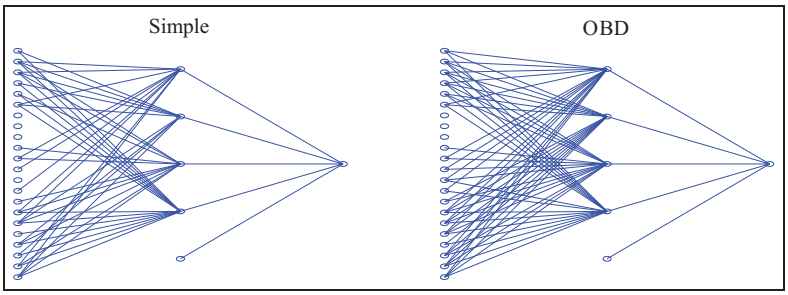


Figure 13. Optimal ANN in test during pruning: simple procedure after 45 pruned weights and OBD after 29 pruned weights.

procedures. It is seen that the OBS network looses accuracy fairly quickly, while the simple method and the OBD maintains a high degree of accuracy even after 70 weights have been deleted.

As the ultimate purpose of the hybrid simulation method is to reduce the computational effort associated with, for example, fatigue analysis of the offshore structure, a robust measure for comparison of the performance of the pruned networks is to evaluate the accuracy with respect to summation of stress cycles using the rain flow counting method.[16] Thus, the comparison is not based on a complete fatigue life analysis including S-N curves and so on but merely based on a summation of stress cycles. The summation is performed for simulations of all seven sea states, and deviations relative to the full RIFLEX simulations are listed in Table 1. This table shows results at different stages of pruning for all three methods, and the values confirm the results from Figure 8. The simple and the OBD procedures are able to maintain a small deviation much longer into the pruning procedure than the corresponding OBS procedure. Furthermore, the last column of this table shows the accuracy of the optimal networks found by the three procedures. For each step in the pruning procedures, the test error is measured and after the pruning, the best network in test is selected as the "optimal." Again, the simple method and the OBD perform superior to the OBS and reach the optimized networks much later in the pruning process. The architectures of the final optimized ANNs obtained by the simple pruning procedure and OBD are indicated in Figure 13. Although the two methods reach their optimum at different stages of the pruning and do not identify the exact same weights to be pruned during the process, the two algorithms clearly identify the same groups of weights as being the least important.

The training of the ANN initiates from a set of random weights. As the gradient descent technique used for minimizing the error during the training does not guarantee that the global minimum is reached, two networks trained with the same set of data will possibly have different weights in the final configuration. This means that the starting point for the pruning algorithms will be different if the entire procedure is repeated. Thus, the optimization procedures discussed in this article will find different orders of pruning. However, it is found that the overall pattern with respect to increase in error as the pruning algorithms progress is the same when the procedure is repeated. The level of success for the three applied pruning methodologies seems to contradict what has been stated in many textbooks and articles.[8,11] The very simple approach of assuming that small weights are less important than large weights works surprisingly well and designates the same weights for pruning as the OBD. The success of the simple approach is presumably a consequence of the data normalization which provides that all input signals have the same order of magnitude. Thereby, the saliency of the individual weights becomes a direct part of the training, as the ANN is able to identify the least important input variables and therefore reduces the magnitude of the corresponding network weights. The OBS procedure, on the other hand, performs very poorly in the given example. The trained ANN used in this example yields a Hessian matrix with very small diagonal elements. Thus, inversion of this matrix leads to numerical problems, which is why the procedure breaks down unless this problem is addressed. The small diagonal elements indicate that the corresponding weights have

little significance with respect to the ANN output. One way to avoid the very small diagonal elements is by network regularization which effectively adds the term $\alpha|W|^2$ to the error function given in equation (2). Regularization can mitigate the numerical problems associated with the inversion of the Hessian matrix. However, regularization is usually applied in order to prevent the network from overfitting to noise in the training data. In the present example where the ANN is trained to simulate the exact solution to the equation of motion, there is no noise in the training data and that may be the reason why the OBS algorithm performs poorly in this specific case.

Concluding remarks

It has been demonstrated how an ANN which has already undergone an overall optimization can be reduced further in size by different optimization procedures. Using the most simple approach and the OBD procedure, it has been possible to prune more than half of the original network weights and still maintain a very high simulation accuracy. The literature[8] indicates that the OBS is superior to the OBD when tested on the classical XOR problem[17] and the MONK's problem.[18] However, in the present case, where the aim has been to obtain the dynamic response of a mechanical structure, the simple and the OBD procedures turned out to be more effective than the OBS with respect to the degree of pruning. After fairly few network weights are pruned by the OBS procedure, the error begins to increase beyond an acceptable level, while when using the other two algorithms, very robust and reliable procedures have been obtained, which are able to effectively identify insignificant inputs. One of the apparent advantages of the OBS is that the adjustment of weights is done without retraining of the ANN. However, since it is sufficient to use relatively few iterations in this retraining part, the additional computational effort associated with OBD is very limited. Thus, the OBD procedure seems to be very effective for the marine structure ANN application considered in this article. As mentioned in section "Network design," the ANN used in this study has been optimized in terms of model memory and size of hidden layer before the optimization algorithms are applied. This initial optimization maybe can be superseded if the optimization algorithms are applied to an "oversize" ANN. However, as networks with too many elements do not generalize well,[7] this may not be a feasible approach. This topic has not been investigated in this study.

Declaration of conflicting interests

The authors declare that there is no conflict of interest.

Funding

This work has been partly funded by the Danish Agency for Science and Innovation as part of the Industrial PhD program.

References

1. Adeli H. Neural networks in civil engineering: 1989–2000. *Comput-Aided Civ Inf* 2001; 16: 126–142.
2. Bakhary N, Hao H and Deeks AJ. Damage detection using artificial neural network with consideration of uncertainties. *Eng Struct* 2007; 29: 2806–2815.
3. Guarize R, Matos NAF, Sagrilo LVS, et al. Neural networks in dynamic response analysis of slender marine structures. *Appl Ocean Res* 2008; 29: 191–198.
4. Christiansen NH, Voie PET, Sødahl N, et al. Efficient mooring line fatigue analysis using a hybrid method time domain simulation scheme. In: *Proceedings of the OMAE2013 international conference on ocean, offshore and arctic engineering*, Nantes, 9–14 June 2013, vol. 32, paper no. 10682. New York: ASME.
5. Christiansen NH, Voie PET, Winther O, et al. Comparison of error measures for neural networks for simulation of slender marine structures. *J Appl Math* 2014; 2014: 759834.
6. Reed R. Pruning algorithms—a survey. *IEEE T Neural Networ* 1993; 4: 740–747.
7. Le Cun Y, Denker JS and Solla SA. Optimal brain damage. In: Touretzky DS (ed.) *Advances in neural information processing systems*, vol. 2. San Mateo, CA: Morgan Kaufmann, 1990, pp.598–605.
8. Hassibi B, Stork DG and Wolff GJ. Optimal brain surgeon and general network pruning. *IEEE IJCNN* 1993; 1–3: 293–299.
9. Warner B and Misra M. Understanding neural networks as statistical tools. *Am Stat* 1996; 50(4): 284–293.
10. Bishop CM. *Pattern recognition and machine learning*. New York: Springer, 2006.
11. Bishop CM. *Neural networks for pattern recognition*. Oxford: Clarendon Press, 1995.
12. Hanson SJ and Burr DJ. Minkowski-r back-propagation: learning in connectionist models with non-Euclidean error signals. In: *Proceedings of IEEE Conference on Neural information processing systems – Natural and Synthetic* (ed DZ Anderson), Denver, CO, 8–12 November 1987, pp. 348–357. New York: American Institute of Physics.
13. Fletcher R. *Practical methods of optimization*. Hoboken, NJ: Wiley, 1987.
14. MARINTEK. *SIMO theory manual*. Trondheim: Sintef, 2009.
15. MARINTEK. *RIFLEX theory manual*. Trondheim: Sintef, 2008.
16. Dowling SD and Socie DF. Simple rainflow counting algorithms. *Int J Fatigue* 1982; 4: 31–40.
17. McClelland JL. Explorations in parallel distributed processing: a handbook models, programs, and exercises, February 2013. California, USA: Stanford University.
18. Thrun SB, Bala J, Bloedorn E, et al. *The monk's problems—a performance comparison of different learning algorithms*. Technical report, CMU-CS-91-197, 1991. Pittsburgh, PA: Carnegie Mellon University.

Paper[P4]

Optimized Mooring Line Simulation Using a Hybrid Method Time Domain Scheme

N. H. Christiansen, P. E. T. Voie, J. Høgsberg, N. Sødahl

Proceedings of the OMAE 2014 International Conference on Ocean, Offshore and Arctic Engineering

Proceedings of the OMAE 2014 International Conference on Ocean, Offshore and Arctic Engineering
OMAE 2014
June 8-13, 2014, San Francisco, USA

OMAE2014-23939

OPTIMIZED MOORING LINE SIMULATION USING A HYBRID METHOD TIME DOMAIN SCHEME

Niels Hørbye Christiansen*
Structures, Pipeline and Risk
DNV Danmark
2900 Hellerup
Denmark
Email: Niels.Christiansen@dnv.com

Per Erlend Torbergsen Voie
Technical Advisory
Det Norske Veritas
7496 Trondheim
Norway
Email: Per.Erlend.Voie@dnv.com

Jan Høgsberg
Department of Mechanical Engineering
Technical University of Denmark
2800 Kgs. Lyngby
Denmark
Email: jhg@mek.dtu.dk

Nils Sødahl
Riser Technology
Det Norske Veritas
1322 Høvik
Norway
Email: Nils.Sodahl@dnv.com

ABSTRACT

Dynamic analyses of slender marine structures are computationally expensive. Recently it has been shown how a hybrid method which combines FEM models and artificial neural networks (ANN) can be used to reduce the computation time spend on the time domain simulations associated with fatigue analysis of mooring lines by two orders of magnitude. The present study shows how an ANN trained to perform nonlinear dynamic response simulation can be optimized using a method known as optimal brain damage (OBD) and thereby be used to rank the importance of all analysis input. Both the training and the optimization of the ANN are based on one short time domain simulation sequence generated by a FEM model of the structure. This means that it is possible to evaluate the importance of input parameters based on this single simulation only. The method is tested on a numerical model of mooring lines on a floating offshore installation. It is shown that it is possible to estimate the cost of ignoring one or more input variables in an analysis.

INTRODUCTION

In resent years an increasing amount of machine learning techniques has proven to be useful in many aspects of structural engineering. Especially the use of Artificial Neural Networks (ANN) has drawn vast attention over the last decades. Adeli [1] gives a quick overview of the broad variety of neural network applications in civil engineering. Machine learning techniques are mostly used to extract information from large data sets. Very often it is a matter of identifying patterns in the data. This ability to recognize patterns can be very useful in damage detection on existing structures. In that case the application detects changes in structural behavior which can be consequences of damages in structural members, see eg. Bakhary et al. [2].

Also in structural design analysis neural networks hold a great potential as these are capable of performing nonlinear analysis at a remarkably high pace, see e.g. Ordaz-Hernandez et. al. [3] who demonstrated how ANN can be trained to predict the deflection of a nonlinear cantilevered beam. This approach has also proven to be very use full in dynamic analysis. Guar-

*Address all correspondence to this author.

ize et. al. [4] showed how a hybrid method combining the Finite Element Method (FEM) and an ANN can simulate the dynamic response of a flexible oil pipe in service about 20 times faster than ordinary nonlinear numerical analysis. Christiansen et al. [5] used a similar approach to conduct a full fatigue life analysis of a mooring line on a floating platform cutting down the simulation time dramatically. The same platform structure was used in [6] where different error functions used during network training were tested on a very small and compact ANN. The trained network used in that analysis simulated more than two orders of magnitude faster than a corresponding numerical FEM analysis.

Recently it has been shown in [7] how the ANN designed in [6] can be further optimized and reduced in size. The literature suggest various kinds of optimization schemes [8], where the study in [7] considers two of the most renown pruning methods for optimizing ANN architecture: Optimal Brain Damage (OBD) [9] and Optimal Brain Surgeon (OBS) [10]. For the specific application in [7] the OBD procedure performed far superior to OBS and it was able to delete more than half of the network connections and still maintain its simulation accuracy.

The present paper demonstrates how the optimization operations can be used as a quick ranking of the importance of the various inputs that go into the analysis. Dynamic response of slender marine structures, such as the mooring lines on the floating platform, always depends on several external loads. By applying the pruning procedure to a trained ANN it is possible to determine which of these external load components that play the least significant role and also evaluate if one or more components can be ignored completely in the analysis. Hence, the optimization procedures can serve as an input parameter study based on one simulation only. In the example presented in this paper it is possible to completely exclude one of the input variables and thereby actually improve the accuracy of the simulations significantly. The results are compared with respect to accumulated structural fatigue damage.

1 Artificial Neural Networks

The artificial neural network is a machine learning tool with a simple brain-like structure that makes it capable for recognizing patterns in large data sets. This ability can be utilized to do non-linear simulation at a very high pace. See e.g. Warner and Misra [11] for a fast thorough introduction to neural networks and their features. In the following an ANN which has been trained to recognize and predict the relationship between the motion of a floating platform and the resulting tension forces in a selected mooring line will be optimized by use of the OBD pruning procedure . The ANN was designed and trained in [6].

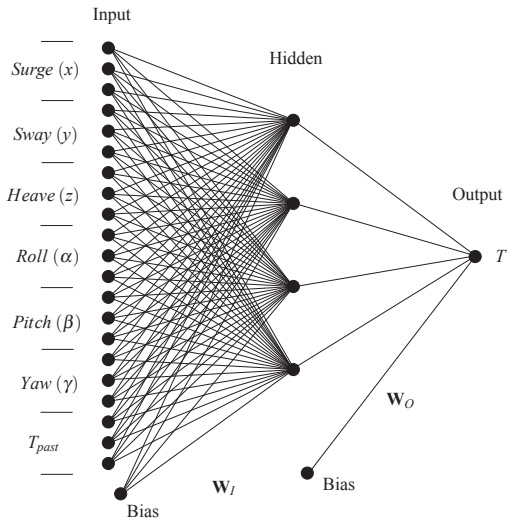


FIGURE 1. Sketch of artificial neural network for predicting top tension force in mooring line.

1.1 Architecture

The architecture of a typical one layer artificial neural network is shown in Figure 1. The ANN consists of an input layer, where each input neuron represents a measured time discrete state of the system. In the present case in Figure 1 the neurons of the input layer represent the six motion components $(x, y, \dots)$ of the floating platform and the previous time discrete mooring line tension force (T_{past}). The input layer is connected to a single hidden layer, which is then connected to the output layer representing the tension force (T) in the selected mooring line. Two neurons in neighboring layers are connected and each of these connections has a weight. The training of an ANN corresponds to an optimization of these weights with respect to a particular data training set. The accuracy and efficiency of the network depends on the network architecture, the optimization of the individual weights and the choice of error function used in the applied optimization procedure.

The design and architecture of the ANN, and the subsequent training procedure, follow the approach outlined in [12]. Assume that the vectors $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ contain the neuron variables of the input layer, output layer and hidden layer, respectively. The output layer and hidden layer values can be calculated by the expressions

$$\mathbf{y} = \mathbf{W}_O^\top \mathbf{z}, \qquad \mathbf{z} = \tanh\left(\mathbf{W}_I^\top \mathbf{x}\right), \qquad x_0 \equiv z_0 \equiv 1, \quad (1)$$

where $\mathbf{W}_I$ and $\mathbf{W}_O$ are arrays that contain the neuron connection weights between the input and the hidden layer and the hidden and the output layer, respectively. By setting x_0 and z_0 perma-

nently to one, biases in the data can be absorbed into the input and hidden layer. The tangent hyperbolic function is used as an activation function between the input and the hidden layer. A non-linear activation function is needed in order to introduce non-linearities into the neural network. The tangent hyperbolic is often used in networks performing regression between continuous variables, because it provides fast convergence in the network training procedure, see [13].

The optimal weight components of $\mathbf{W}_I$ and $\mathbf{W}_O$ are found by an iterative procedure, where the weights are modified to give a minimum with respect to a certain error function. The literature suggests many choices of error functions. In this case we use the Minkowski$-R$ error [14] with $R = 4$, which has shown to yield good ANN performance for this particular application [6]. The error function can be written as

$$E(\mathbf{W}) = \frac{1}{4}\sum_{n=1}^{N}\sum_{k=1}^{c}\{y(\mathbf{x}^n;\mathbf{W}_I;\mathbf{W}_O)_k - t_k^n\}^4, \qquad (2)$$

where y is the ANN output and t is a target value that corresponds to a specific input. The updating of the weight components is performed by a classic gradient decent technique, which adjusts the weights in the opposite direction of the gradient of the error function [15]. For the ANN this gradient decent updating can be written as

$$\mathbf{W}_{new} = \mathbf{W}_{old} + \Delta\mathbf{W}, \qquad \Delta\mathbf{W} = -\eta\frac{\partial E(\mathbf{W})}{\partial \mathbf{W}}, \qquad (3)$$

where η is the learning step size parameter. This parameter can either be constant or updated during the training of the ANN. For the applications in the present paper the learning step size parameter is dynamic and will be adjusted for each iteration, so that it is increased if the training error is decreased compared to previous iteration step and reduced if the error increases. Note that $\Delta\mathbf{W}_I$ and $\Delta\mathbf{W}_O$ must be evaluated separately.

1.2 Optimization Algorithm

The optimization procedure initiates from a reasonably large network that has been trained to minimum error. Even though the ANN used here already is as compact as possible in terms of size of input and hidden layer, it is assumed that the network is still sufficiently large to be pruned further on individual weight level.

In the following the OBD algorithm will be implemented and tested. The idea of the method is to rank the importance of all network weights and then successively delete the least salient weight. The starting point is a Taylor expansion of the error function (2) with respect to network weights. The Taylor series can be written as

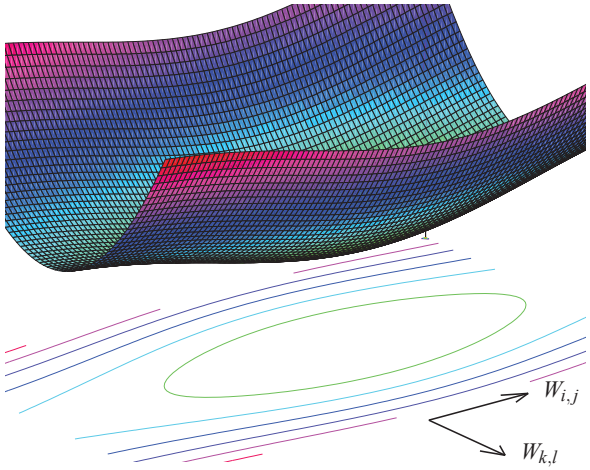


FIGURE 2. The error function around a local minimum. The curvature of the error surface in each direction corresponds to the saliency of the associated weight. In this illustration we see that $\frac{\partial^2 E}{\partial W_{i,j}^2} < \frac{\partial^2 E}{\partial W_{k,l}^2}$.

$$\delta E = (\frac{\partial E}{\partial \mathbf{W}})^T\delta\mathbf{W} + \frac{1}{2}\delta\mathbf{W}^T\mathbf{H}\delta\mathbf{W} + O(\|\delta\mathbf{W}\|^3), \qquad (4)$$

where $\mathbf{H} \equiv \partial^2 E/\partial\mathbf{W}^2$ is the Hessian matrix containing all second order derivatives. The OBD optimization method is based on networks that are trained to a local minimum. This means that the first term in (4) vanishes, $(\delta E/\delta\mathbf{w} = \mathbf{0})$. Furthermore, higher order terms are ignored. Thereby the saliency of the network weights is calculated merely by looking at the second order derivatives. Thus, the problem reduces to

$$\delta E \approx \frac{1}{2}\delta\mathbf{W}^T\cdot\mathbf{H}\cdot\delta\mathbf{W} \qquad (5)$$

This expression gives the curvature in error space at the local minimum that was reached during the network training. The smallest component in the Hessian identifies which weight component that is least significant with respect to increase in error and hence, which weight it is cheapest to delete, as sketched for two dimensions in Figure 2 where each weight represent a dimension in error space.

The OBD procedure trains the initial ANN configuration, which is fully connected between inputs and hidden layer and between hidden layer and outputs. The saliencies of all weights are computed and the least salient weight is pruned (put perma-

nently to zero). After each time a network weight is deleted the network must be retrained before the Hessian can be reevaluated and the next least salient weight can be selected. The procedure can in principal be repeated until the network is completely gone. Each time a weight has been pruned the network is tested and after the pruning the best network in the test is selected.

The procedure approximates the Hessian by disregarding the non-diagonal terms. With the error function given in (2) the assumption of a diagonal Hessian leads to the following second order derivatives of the network error with respect to the two weight matrices $\mathbf{W}_O$ and $\mathbf{W}_I$, respectively

$$\frac{\partial^2 E}{\partial W_{O,j}^2} = 3\sum_{n=1}^{N}(y_n - t_n)^2 z_j^2 \tag{6}$$

$$\frac{\partial^2 E}{\partial W_{I,ji}^2} = \sum_{n=1}^{N}(1-z_j^2)\left(3(1-z_j^2)\cdot(y_n-t_n)^2 W_{0,ji} - 2z_j(y_n-t_n)^3\right) W_{0,ji}x_i^2 \tag{7}$$

where N is the number of training data sets. Equation (6) and (7) give the saliency for all network weights whereby the least salient weight can be identified and deleted.

2 Application to marine structure

The structure that the ANN is trained to simulate is sketched in Figure 3. It consists of a floating offshore platform located at 105 m water depth, which is anchored by 18 mooring lines assembled in four main clusters. The external forces acting on the structure are induced by waves, current and wind.

2.1 Structural Model

In principal the dynamic analysis of the platform-mooring system corresponds to solving the equation of motion

$$\mathbf{M}(\mathbf{r})\ddot{\mathbf{r}} + \mathbf{C}(\mathbf{r})\dot{\mathbf{r}} + \mathbf{K}(\mathbf{r})\mathbf{r} = \mathbf{f}(t), \tag{8}$$

In this nonlinear equation $\mathbf{r}$ contains the degrees of freedom of the structural model, $\mathbf{f}$ includes all external forces acting on the structure from for example gravity, buoyancy and hydrodynamic effects, while the non-constant matrices $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{K}$ represent the system inertia, damping and stiffness, respectively. The system inertia matrix $\mathbf{M}$ takes into account both the structural inertia and the response dependent hydrodynamic added mass. Linear and nonlinear energy dissipation from both internal structural damping and hydrodynamic damping are accounted for by the damping matrix $\mathbf{C}$. Finally, the stiffness matrix contains contributions from both the elastic stiffness and the response dependent geometric stiffness.

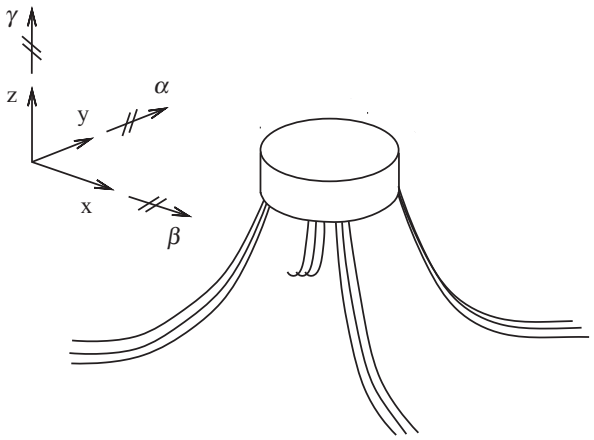


FIGURE 3. Mooring system with floating platform and anchor lines.

The nonlinear equations of motion in (8) couple the structural response of the floating platform and the response of the mooring lines. However, the system is effectively solved by separating the solution procedure into the following steps. First the motion of the floating platform is computed by the program SIMO [16], assuming a quasi-static catenary mooring line model with geometric nonlinearities. The platform response from this initial analysis is subsequently used as excitation in terms of prescribed platform motion in a detailed nonlinear finite element analysis for the specific mooring line with highest tension stresses. The location of the mooring line with largest stresses is indicated in Figure 4. For this specific line the hot-spot with respect to fatigue is located close to the platform, and is in the following referred to as the top tension force. From the detailed fully nonlinear analysis performed by RIFLEX [17] the time history of the top tension force at this hot-spot is extracted.

In [6] an ANN was trained to predict the top tension forces in the selected mooring line with the platform response variables as network input. The network training was based on the simulated time histories for both the platform motion and the top tension force. In [5] a multi layer ANN was trained to simulate the top tension force two orders of magnitude faster than a corresponding numerical model. The training data was set up and arranged so that a simple ANN with a single hidden layer could simulate all fatigue relevant sea states and thereby provide a significant reduction in the computational effort associated with a fatigue life evaluation. For clarity the ANN used in [6] covered only a few sea states, with different significant wave heights and constant peak period. This gave a compact neural network that is conveniently used in this paper to illustrate the influence of

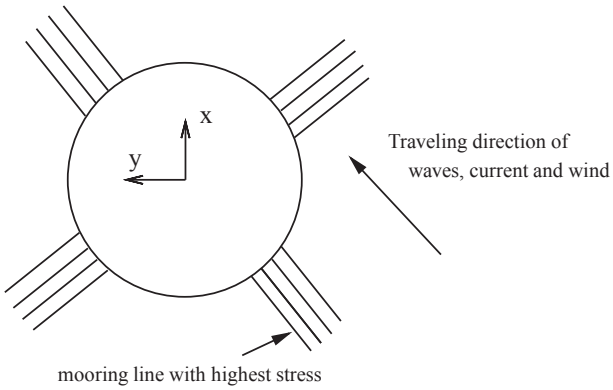


FIGURE 4. Mooring line system (top view).

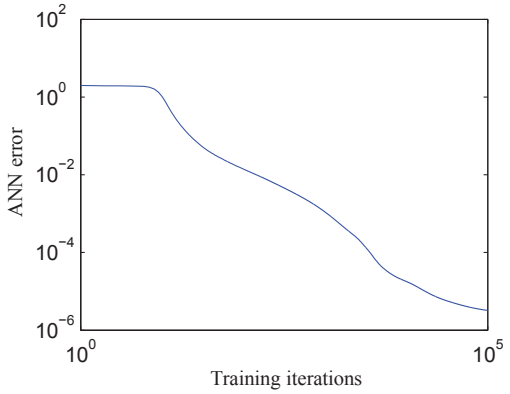


FIGURE 7. Tension force history used as ANN training target data.

applying the optimization algorithm.

2.2 Network design

Optimal network architecture and design for this application was investigated in [6]. It was shown that a network with four neurons in the hidden layer and a model memory of three time steps gave a small and very accurate ANN. Using the platform motion together with the top tension force of previous time steps as ANN input gives an input vector which can be written as

$$\mathbf{x}_n = \Big[[x_t \quad x_{t-h} \quad x_{t-2h}] \quad [y_t \quad y_{t-h} \quad y_{t-2h}] \quad [z_t \quad z_{t-h} \quad z_{t-2h}] \quad [\alpha_t \quad \alpha_{t-h} \quad \alpha_{t-2h}] \quad [\beta_t \quad \beta_{t-h} \quad \beta_{t-2h}] \quad [\gamma_t \quad \gamma_{t-h} \quad \gamma_{t-2h}] \quad [T_{t-h} \quad T_{t-2h} \quad T_{t-3h}] \Big]^T, \tag{9}$$

where T denotes the top tension force, h is the time increment and x, y, z, α, β and γ represent platform motion (surge, sway, heave, roll, pitch and yaw, respectively). The corresponding ANN output is the current top tension force T_t in the mooring line

$$y_n = T_t \tag{10}$$

Since there is only one network output y is now a scalar and not a vector as in (1). For the training of the ANN nonlinear simulations in RIFLEX are conducted for sea states with a significant wave height of $H_s = 2$ m, 8 m and 14 m, respectively. With these wave heights included in the training data the ANN is able to provide accurate time histories for the top tension force for all intermediate wave heights [6]. The data that is used for training of the ANN is shown in Figures 5 and 6. Figure 5 shows the time histories for the six motion degrees of freedom of the platform calculated by the initial analysis in SIMO and used as input to both the full numerical analysis in RIFLEX and the ANN training and simulation. Figure 6 shows corresponding time history for the top tension force determined by RIFLEX. The full time histories shown in Figures 5 and 6 are constructed from time series for the three significant wave heights. The first 830 seconds of the training data set represent 2 m significant wave height, the next 830 seconds are for 8 m wave height and the remaining part is then for 14 m wave height.

2.3 Network training

As mentioned previously it is one of the basic assumptions for the optimization method that the pruning procedure initiates from a network which has been trained to a local minimum. Hence, the first step when applying the method is to train the ANN. Figure 7 shows the development in ANN error during training. It is assumed that 10^5 training iterations are sufficient to consider the network as having reached a local minimum in error. Note that the ANN error is plotted on logarithmic axes. The full network and the network simulation compared to the RIFLEX simulation are shown in Figure 8.

The optimization algorithm has been applied to the trained ANN. Figure 9 shows the development of ANN error as the number of deleted network weights increases. To verify that the procedure actually selects and deletes appropriate weights an additional algorithm that simply deletes random weights is also included in the graph. The retraining of the ANN applies for both algorithms in the graph.

The initial network training requires a fairly large amount of iterations to reach a local minimum in error. However, the retraining of the network performed after each weight deletion does not have to be as thorough. In this example 2000 iterations in retraining have shown to give satisfactory results, which

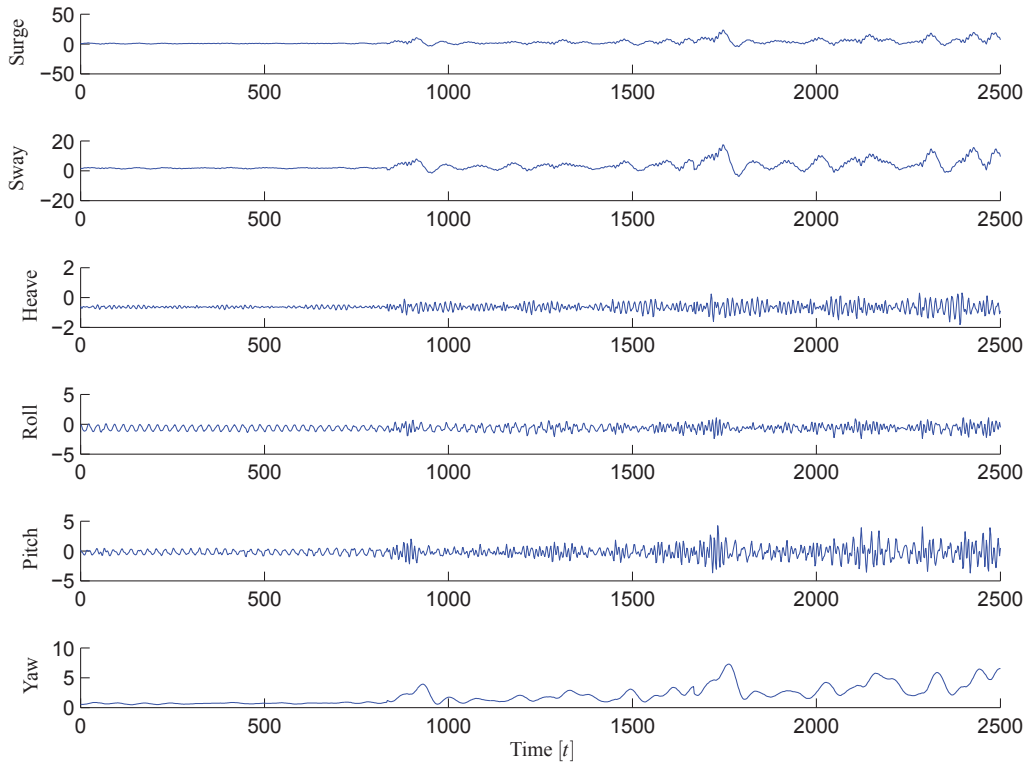


FIGURE 5. Platform motion used as ANN training input data.

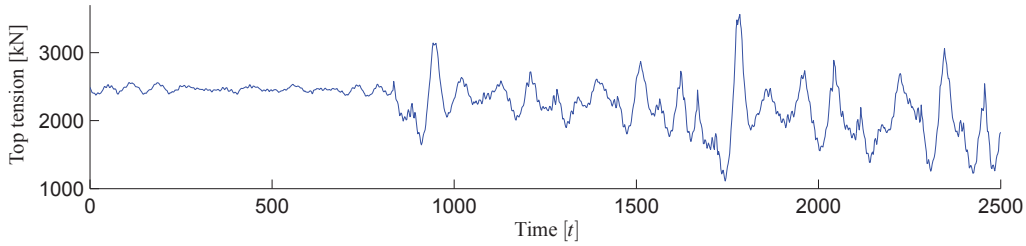


FIGURE 6. Tension force history used as ANN training target data.

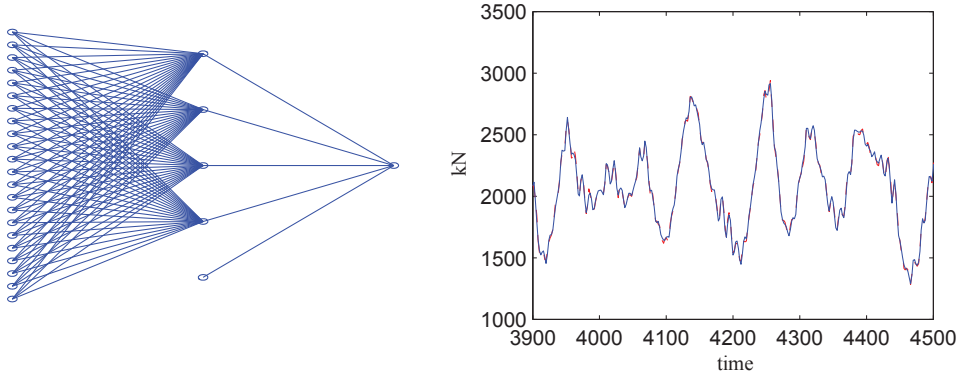


FIGURE 8. Full network before pruning. ——— ANN, – – – RIFLEX

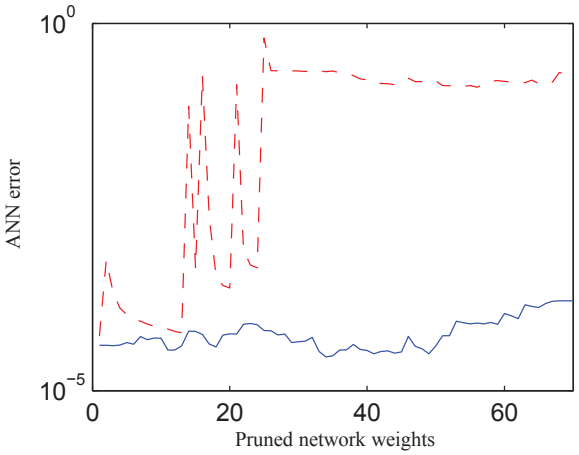


FIGURE 9. Error as function of pruned network weights. —— OBD, – – – Random.

means that the retraining is not nearly as time consuming as one might imagine at first glance at the procedure. Hence, the retraining is just a brief adjustment to the weights and not a complete network training.

2.4 Network optimization

The original purpose of the OBD procedure is to optimize the ANN and identify the best network in test. Figure 9 shows that OBD finds the best ANN after 34 weight deletions. However, as the pruning procedure successively cuts network weights until the entire network is deleted it will at certain stages reach a point where a motion degree of freedom for the platform is completely ignored by the network. This can be used to rank the importance of the individual motion components and perhaps to evaluate weather one or more components are in fact negligible. In this example the optimal network emerges around the same stage as the heave motion of the platform is completely removed from the network input. The heave motion component is completely ruled out after 37 deletions. The heave motion of the platform is represented in the ANN input layer by neurons 7-9, see Figure 1. The ANN with 37 pruned weights is shown in figure 10. The figure also shows the ANN simulation together with the RIFELX series. The fact that the optimal ANN is reached as the heave motion is ruled out indicates that not only is the heave motion redundant it actually adds noise to the system. The explanation for this is that the numerical model has a build-in heave compensator that dampens the effect of the vertical platform motion. Hence, the tension forces in the mooring line is not directly related to the heave motion of the floating platform.

The next degree of freedom to be completely ignored by the network is the platform roll motion. This happens after 53 deletions. The network and corresponding simulation performance is plotted in figure 11 (The roll motion of the platform is represented in the ANN input layer by neurons 10-12, see Figure 1). After 74 deletions the third motion degree of freedom is ignored by the network. At this stage it is the platform pitch motion which no longer plays a part in the ANN simulation (The pitch motion of the platform is represented in the ANN input layer by neuron 13-15). Figure 12 shows the network at this stage and also that it apparently still simulates the tension force with high accuracy. Figure 13, on the other hand, clearly shows that continuing the pruning until a fourth motion component (surge) is deleted is going to far. At this stage the network still captures the overall dynamics of the structure, but it is no longer able to do any useful simulation.

It is difficult to relate the error measure in Figure 9 to how accurate the ANN actually simulates. Visual inspections of the curves in Figures 10-13 give an indication of how well the networks perform. However, as the ultimate purpose of the hybrid method is to reduce the computational effort associated with fatigue analysis, an obvious measure for comparing the performance of the pruned networks is to evaluate their accuracy on summarized stress cycles using rain flow counting. The summation is done for all seven sea state simulations. Deviations from the RIFLEX simulation are listed in Table 1. The table lists results at different stages of pruning. The numbers confirm the graph in Figure 9. The highest accuracy is obtained by the network that ignores the heave motion but includes the rest of the platform motion. This ANN lies very close to the expected optimal network at 34 deleted weights.

The training of the ANN initiates from a set of random weights. As the gradient descent technique used for minimizing the error during the training does not guarantee that the global minimum is reached, two networks trained on the same set of data will most likely have completely different weights in the final configuration. This means that the starting point for the pruning algorithms and also the pruning sequence will change if the procedure is repeated. However, the overall pattern with respect to increase in error and which weights are deleted as the pruning algorithms progress is the same when the procedure is repeated.

To test how well the OBD works as a tool for quick parameter study, a series of RIFLEX simulations are conducted in accordance with the ranking given by the OBD procedure. Hence, the platform heave motion is left out of the first calculation, heave and roll left out in the second calculation and so forth. The deviations in accumulated stress cycles are listed in Table 2. Corresponding time domain simulations are shown in Figure 14. It is seen that leaving out the heave motion does not degrade the accuracy of the simulation significantly. When leaving out heave and roll motion from the analysis, the error increases slightly - as seen for the ANN simulation. For some reason, when the pitch motion is ignored the error in the stress cycle accumulation drops to practically nothing. This is in contrast to the ANN simulations where we saw a general increase in error as the number of input

TABLE 1. Deviations on accumulated tension force cycles when compared to RIFLEX simulations.

Simulation	ANN	ANN_1	ANN_2	ANN_3	ANN_4
Deviations	1.2 %	−1.1 %	−3.4 %	−8.8 %	7.7 %

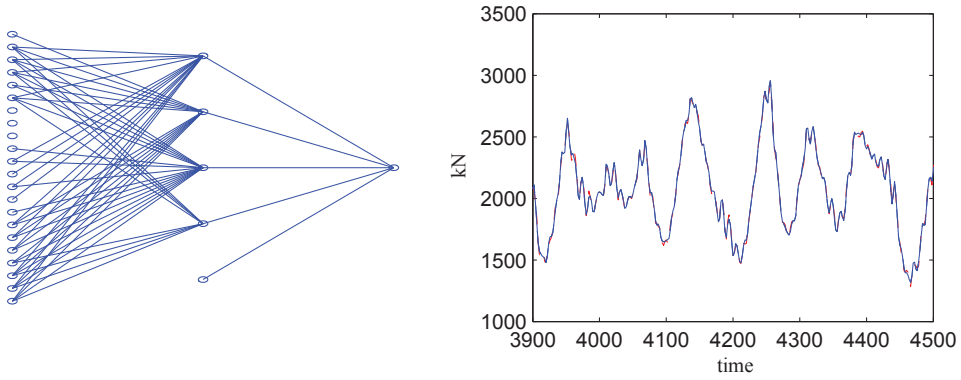


FIGURE 10. ANN_1: Network with one input component pruned (heave) - 37 weights deleted by OBD ——— ANN, – – – RIFLEX

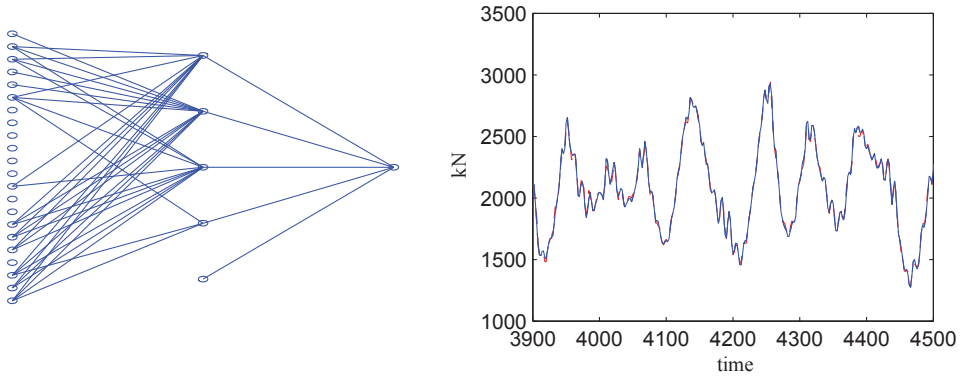


FIGURE 11. ANN_2: Network with two input components pruned (heave and roll) - 53 weights deleted by OBD. ——— ANN, – – – RIFLEX

TABLE 2. Deviations on accumulated tension force cycles when compared to RIFLEX simulations.

Simulation	RIFLEX	$RIFLEX_1$	$RIFLEX_2$	$RIFLEX_3$	$RIFLEX_4$
Deviations	-	-0.5 %	3.1 %	0.03 %	-65.4 %

variables were ignored. Nothing in the simulation comparison in Figure 14c indicates that this analysis is more accurate than the previous. A closer look at the data (not included in this paper) reveals that the surprisingly low deviation in accumulated load stress cycles is coincidental. When all four input variables are ignored (heave, pitch, roll and surge) the simulation is completely useless. The inaccuracy of accumulated load stress cycles increases dramatically and simulation in Figure 14d is obviously way off.

Except for the very low error in the fatigue estimation based on the simulation conducted with RIFLEX ignoring heave, pitch and roll, the results obtained by the two methods coincides very well. Based on just one relatively short simulation sequence including three different sea states, the trained ANN combined with the OBD procedure is able to evaluate the importance of the input variables and to estimate the cost of ignoring one or more variables in the analysis including seven different sea states. In addition, the pruning procedure detects the stage where the sim-

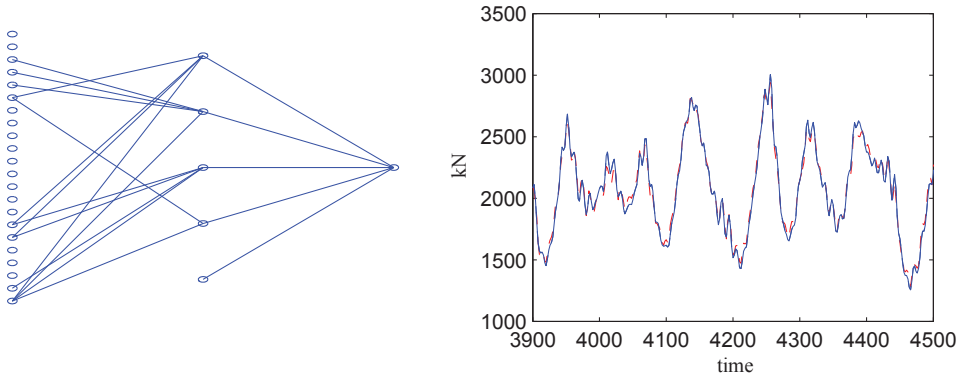


FIGURE 12. ANN_3: Network with two input components pruned (heave, roll and pitch) - 74 weights deleted by OBD. ——— ANN, – – – RIFLEX

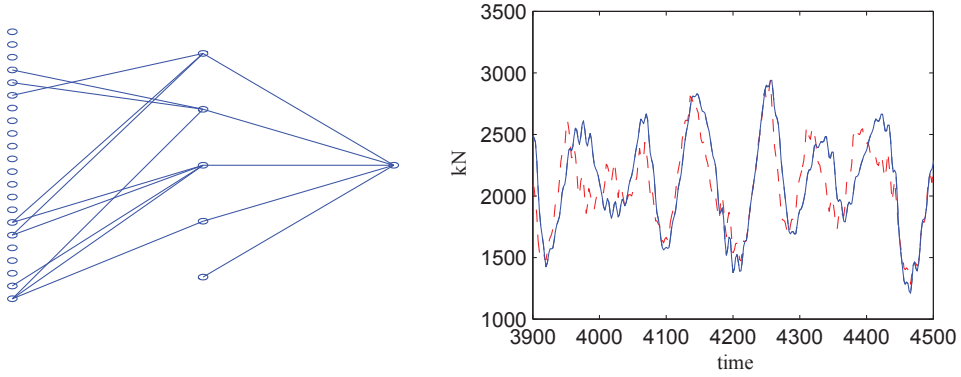


FIGURE 13. ANN_4: Network with four input components pruned (heave, roll, pitch and surge) - 77 weights deleted by OBD. ——— ANN, – – – RIFLEX

ulation becomes completely useless. Noting that training and pruning an ANN is a lot less time consuming than running several numerical simulations, the describe procedure can become a very valuable tool when investigating the importance of various input variables in mechanical analysis.

3 Concluding remarks

It has been shown how an ANN which has already undergone an overall optimization can be reduced further in size by the OBD optimization procedure. Using the OBD procedure it was possible to prune more than half of the original network weights and still maintain a very high simulation accuracy. In this process it turned out that three out of six platform motion components could be completely ignored in the simulation input without significantly impairing the accuracy of the ANN. In fact, leaving out one of the input variables improved the simulation accuracy of the ANN. The results were tested on additional RIFLEX simulations where input variables were ignored in accordance with the recommendations given by the OBD algorithm. The additional simulations showed that the ANN/OBD procedure can be used to identify redundant variables and also be used to give an estimate of the cost of simplifying an analysis.

Using OBD seems like a very robust and reliable procedure able to effectively identify insignificant inputs. Hence, this method can be used as a cost effective parameter study based on one simulation sequence only.

REFERENCES

[1] Adeli, H., 2001. "Neural networks in civil engineering: 1989-2000". *Computer-Aided Civil and Infrastructure Engineering,* ***16***, pp. 126–142.

[2] Bakhary, N., Hao, H., and Deeks, A., 2007. "Damage detection using artificial neural network with concideration of uncertainties". *Engineering Structures,* ***29***, pp. 2806–2815.

[3] Ordaz-Hernandez, K., Fisher, X., and Bennis, F., 2008. "Modal reduction technique for mechanical behaviour

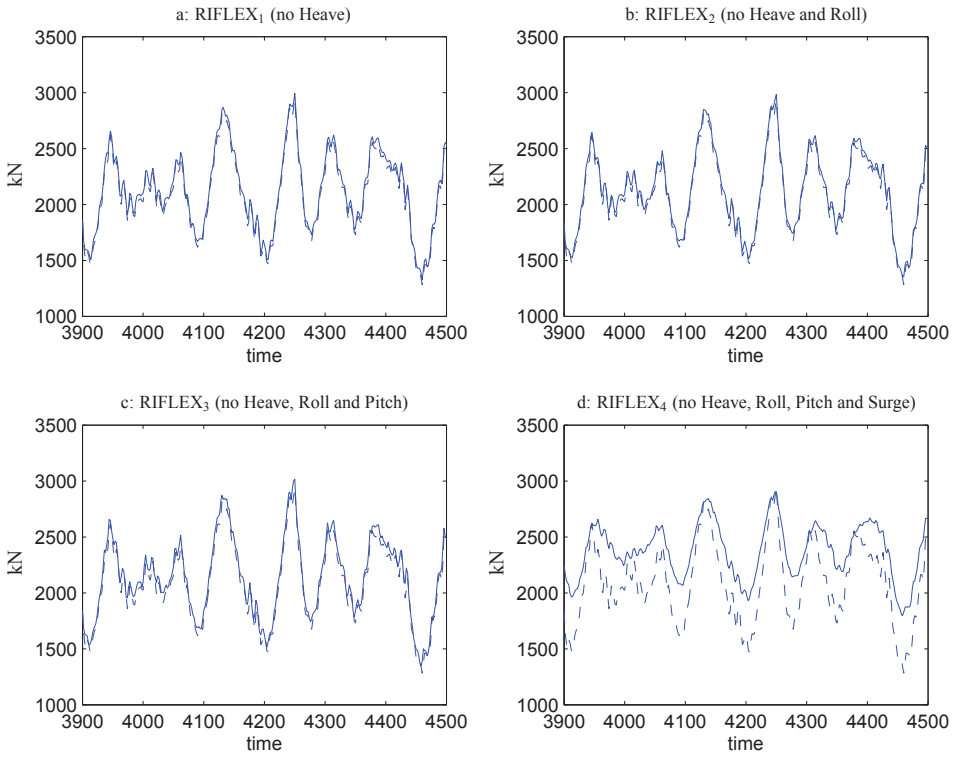


FIGURE 14. RIFLEX simulations. Reduced input compared to original simlation. ——— RIFLEX with reduced input, – – – RIFLEX

modelling: Efficiency criteria and validity domain assessment". *Journal of Mechanical Engineering Science,* ***222***, pp. 493–505.

[4] Guarize, R., Matos, N., Sagrilo, L., and Lima, E., 2008. "Neural networks in dynamic response analysis of slender marine structures". *Applied Ocean Research,* ***29***, pp. 191–198.

[5] Christiansen, N., Voie, P., Sødahl, N., and Høgsberg, J. "Efficient mooring line fatigue analysis using a hybrid method time domain simulation scheme". *Procedings of the OMAE2013 International Conference on Ocean, Offshore and Arctic Engineering*.

[6] Christiansen, N., Voie, P., Winther, O., and Høgsberg, J. "Comparison of neural network error measures for simulation of slender marine structures". *TBA*.

[7] Christiansen, N., Voie, P., Winther, O., and Høgsberg, J. "Optimization of neural networks for simulation of slender marine structures". *TBA*.

[8] Reed, R., 1993. "Pruning algorithms - a survey". *IEEE Transactions on Neural Networks,* ***4***, pp. 740–747.

[9] Cun, Y. L., Denker, J., and Solla, S., 1990. "Optimal brain damage". *In D.S. Touretzky (Ed.), Advances in Neural Information Processing Systems,* ***2***, pp. 598–605.

[10] Hassibi, B., Stork, D., and Wolff, G., 1993. "Optimal brain damage and general network pruning". *1993 IEEE International Conference on Neural Networks, vols 1-3, 293-299*.

[11] Warner, B., and Misra, M., 1996. "Understanding neural networks as statistical tools". *The Amarican Statitiscian,* ***50:4***, pp. 284–293.

[12] Bishop, C. M., 2006. *Pattern Recognition and Machine Learning*. Springer, New York.

[13] Bishop, C. M., 1995. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford.

[14] Hanson, S., and Burr, D., 1987. "Minkowski-r backpropagation: Learning in connectionist models with non-euclidean error signals". In *Neural Information Processing Systems*, D. Anderson, ed. Denver, American Institute of Physics, New York, pp. 348–357.

[15] Fletcher, R., 1987. *Practical Methods of Opimization*. Wiley.

[16] MARINTEK, 2009. *Simo theory manual*. Norway, Trondheim, Sintef.

[17] MARINTEK, 2008. *Riflex theory manual*. Norway, Trondheim, Sintef.

Paper[P5]

Sequential Neural Network Procedure for Nonlinear Time Domain Simulation of Offshore Risers

N. H. Christiansen & J. Høgsberg

Submitted August 2014

Vol. 00, No. 00, Month 20XX, 1–17

Sequential Neural Network Procedure for Nonlinear Time Domain Simulation of Offshore Risers

N.H. Christiansen[a,b*] and J. Høgsberg[b]

[a]*DNV GL Denmark A/S, Tuborg Parkvej 8, 2900 Hellerup, Denmark*; [b]*Department of Mechanical Engineering, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark*

(*Received 00 Month 20XX; final version received 00 Month 20XX*)

This paper presents a novel procedure to utilize artificial neural networks (ANN) in dynamic analysis of nonlinear structures. The procedure is based on a hybrid method that combines the finite element method (FEM) and ANN. This hybrid method has been established over the last decade and provides in many cases a significant reduction in the computational effort associated with nonlinear structural analysis. Unfortunately, the method has only been applied to structures with a very direct load-response relation. The present paper therefore proposes a sequential procedure for the analysis of oil pipes and flexible risers, where it is possible to simply step through the structural model, and thereby simulate the response of the entire riser for a given load history. A simplified numerical riser model is used to demonstrate the sequential procedure for two typical flexible riser configurations. It is demonstrated that when using a series of simple ANNs it is possible to simulate response histories at all critical locations of the flexible risers around two orders of magnitude faster compared to conventional numerical analysis.

Keywords: Nonlinear dynamics; slender marine structures; neural networks; flexible risers

1. Introduction

Over the years oil and gas exploration has moved towards more and more harsh environments. In deep and ultra deep water installations the reliability of flexible risers and anchor lines are of paramount importance. Therefore, the design and analysis of these structures draw an increasing amount of attention. Slender offshore structures such as oil/gas risers and mooring systems exhibit large deflections and therefore require nonlinear numerical models for reliable analysis. Furthermore, in design and analysis of flexible risers and mooring systems for floating offshore units, there is a very pronounced need for long time series simulations - both in evaluation of ultimate limit state and of long term fatigue [1, 2]. The combination of the need for detailed nonlinear models and the demand for long time series simulations makes design and analysis of these types of structures very time consuming and costly. Thus, the motivation for developing time saving models and methods is very high. And even though there have been several suggestions on how to avoid this computationally expensive task, see e.g. [3] which discusses the possibilities of reducing the amount of time domain simulations, it still remains a time consuming part of riser design.

*Corresponding author. Email: Niels.Christiansen@dnvgl.com

The use of artificial neural networks (ANN) for reducing computational cost has shown to be useful in various aspects of structural engineering. Adeli [4] gives an overview of applied methods. Many of these applications deal with detection of structural fatigue damage, see e.g. [5], while others use a neural network approach to simulate load histories [6]. The ability of the ANN to perform nonlinear mapping between a given input and a system output makes it capable of performing response predictions without time consuming equilibrium iterations. Hence, a properly trained ANN can save a lot of time in response simulations and thereby reduce the need for the infuriating compromise between model sophistication and computational efficiency. Many of the hybrid method techniques combine the finite element method (FEM) and ANNs. For example, Ordaz-Hernandez et. al. [7] have shown that an ANN can be trained to predict the deflection of a nonlinear cantilevered beam, while Guarize et. al. [8] applied a similar network structure to simulate the dynamic response of a flexible oil pipe in service, thereby reducing calculation time by a factor of about 20. Within the field of slender marine structures the hybrid method has only been used to predict forces in the upper part of risers and mooring lines.

However, in order to really utilize the method it should be possible to predict section forces at all critical locations along the flexible riser. This paper therefore shows how a series of neural networks can be used to perform simulations of the dynamic response of the full flexible riser model. A simplified two-dimensional FEM model representing typical riser configurations are used to demonstrate the hybrid method. The nonlinear FEM model uses a co-rotational beam element formulation similar to the approach presented by Yazdchi and Crisfield [9]. In order for the authors of the present paper to maintain a complete overview of all details within the numerical simulations, instead of using commercial software, both the FEM analysis of the risers and the subsequent ANN training and response simulations are conducted by small (in-house) numerical toolboxes developed in MATLAB. The examples presented in this paper show that when using the proposed sequential hybrid method scheme it is possible to obtain high simulation accuracy for all parts of the riser structures, while reducing the calculation time by around two orders of magnitude.

2. Artificial Neural Network

The artificial neural network (ANN) is a pattern recognition tool that replicates the ability of the human brain to recognize and predict various kinds of patterns. The reader may consult e.g. Warner and Misra [10] for a fast and thorough introduction to neural networks and their features.

2.1. *Setting up the ANN*

The architecture of the ANN used in this paper is shown in Figure 1. The ANN consists of an input layer, a hidden layer and an output layer. All layers consist of a number of neurons. Each neuron in the input layer represent a time discrete state of the system. The neuron in the output layer gives the system response at a specific location for a given input. In connection with flexible riser analysis the variables u_t and r_t represent the prescribed floating platform motion and the corresponding structural riser response at time t, respectively. Structural response at previous time steps are denoted $r_{t\_}$. This is described in more detail in section

3.2. As the neurons in the hidden layer have no physical interpretation, this layer can be of arbitrary size. However, since the ability of the ANN to learn complex patterns is directly related to the size of this layer, it must have sufficient size to comprise all underlaying dynamic characteristics of the system. Two neurons in neighboring layers are connected and each of these connections have a weight. The training of an ANN corresponds to an optimization of these weights with respect to a particular data training set. The accuracy and efficiency of the network depends on the network architecture, the optimization of the individual weights and the choice of error function used in the applied optimization procedure. The design and architecture of the ANN, and the subsequent training procedure, follow the approach outlined in [11]. Assume that the vectors $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ contain the neuron variables of the input layer, output layer and hidden layer, respectively. The output layer and hidden layer values can be calculated by the expressions

$$\mathbf{y} = \mathbf{W}_O^\top \mathbf{z}, \qquad \mathbf{z} = \tanh\left(\mathbf{W}_I^\top \mathbf{x}\right), \qquad x_0 \equiv z_0 \equiv 1, \tag{1}$$

where $\mathbf{W}_I$ and $\mathbf{W}_O$ are arrays that contain the neuron connection weights between the input and the hidden layer and the hidden and the output layer, respectively. By setting x_0 and z_0 permanently to one, biases in the data can be absorbed into the input and hidden layer. The tangent hyperbolic function is used as an activation function between the input and the hidden layer. A non-linear activation function is needed in order to introduce nonlinearities into the neural network. The tangent hyperbolic is often used in networks of this type, which represent a monotonic mapping between continuous variables, because it provides fast convergence in the network training procedure, see [12].

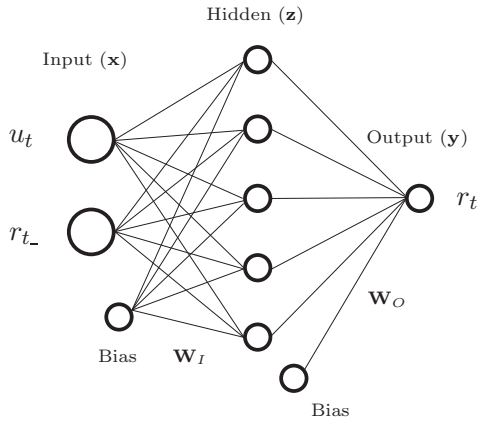


Figure 1. Sketch of artificial neural network.

The optimal weight components of the arrays $\mathbf{W}_I$ and $\mathbf{W}_O$ are found by an iterative procedure, where the weights are modified to give a minimum with respect to a certain error function. The error function which is minimized during training can be written as

$$E\left(\mathbf{W}\right) = \frac{1}{2}\sum_{n=1}^{N}\left\{y(\mathbf{x}_n; \mathbf{W}_I; \mathbf{W_O}) - \tau_n\right\}^2 + \frac{1}{2}\alpha\mathbf{W}^2, \tag{2}$$

where y is the ANN output, N is the number of training data sets, τ is the target

value for a given input $\mathbf{x}$ and $\mathbf{W}$ represents either $\mathbf{W}_I$ or $\mathbf{W}_O$. The weight decay α controls the value of the weights and prevents the ANN from overfiting due to noise in the training data. Since the ANN in this case simulate a theoretical model of a physical system and therefore replicates an exact solution to the system equations of motion there is no noise in the target output data. Thus, the weight decay in (2) is set to zero ($\alpha = 0$) in the following.

The updating of the weight components is performed by a classic gradient decent technique, which adjusts the weights in the opposite direction of the gradient of the error function [13]. For both $\mathbf{W}_I$ and $\mathbf{W}_O$ of the ANN this gradient decent updating can be written as

$$\mathbf{W}_{new} = \mathbf{W}_{old} + \Delta\mathbf{W}, \qquad \Delta\mathbf{W} = -\eta \frac{\partial E\left(\mathbf{W}\right)}{\partial \mathbf{W}}, \tag{3}$$

where η is the learning step size parameter. This parameter can either be constant or updated during the training of the ANN. For this application the dynamic learning step size parameter is adjusted for each iteration so that it is increased if the training error is decreased compared to previous iteration steps and conversely reduced when the training error increases. The two weight arrays $\mathbf{W}_I$ and $\mathbf{W}_O$ must be evaluated separately yielding these weight updates

$$\Delta W_j^O = -\eta \left(\sum_n^N \left(y_n - \tau_n\right) z_j \right) \tag{4}$$

$$\Delta W_{ji}^I = -\eta \left(\sum_n^N \Big(\left(1 - z_j^2\right) W_j^O \left(y_n - \tau_n\right) \Big) x_i \right) \tag{5}$$

The performance of a trained ANN is usually measured in terms of the so-called validation error, which is calculated in the same way as the training error (2) but with respect to an entirely new data set, which has not been part of the network training.

3. Hybrid method

The training of the ANN depends on data that contain the relation between external effects on the structure and the corresponding structural response. This means that the ANN can not stand alone as there must be a source of relevant data. The data can come from measurements, experiments or from numerical models as long as the data contain the dynamic characteristics of the structure in question. In this case, as in most cases, a numerical FEM model will be used as data generator.

3.1. *Numerical Model*

To illustrate the hybrid method a numerical model of a flexible oil/gas pipe is established, as sketched in Figure 2. The FEM model of the structure uses a two-dimensional co-rotational beam element formulation as described in [14]. This formulation is effective for analysis of structures with large deflections because it separates the overall beam motion into two parts: a rigid body type motion associated with a local frame of reference, and a deformation of the beam within this local frame. The local deformations of the beam element can be considered

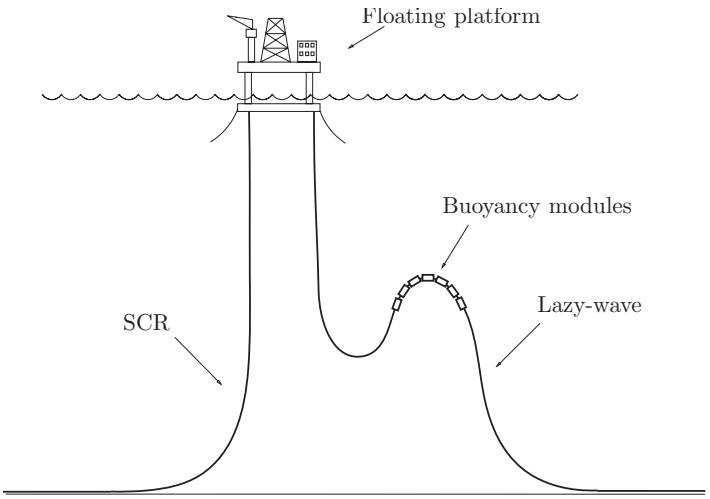


Figure 2. Sketch of two typical riser configurations.

small, but geometrical stiffness contributions which depend on the deformation are introduced. Thus, a nonlinear model which can handle large deflections and deformations is obtained when the governing equations of motion can be stated in the following form,

$$\mathbf{M}\ddot{\mathbf{r}}(t) + \mathbf{C}\dot{\mathbf{r}}(t) + \mathbf{K}(\mathbf{r})\mathbf{r}(t) = \mathbf{f}(t). \tag{6}$$

The lumped masses and beam stiffness contributions are assembled in a mass matrix $\mathbf{M}$ and a (tangent) stiffness matrix $\mathbf{K}$, respectively. Structural damping is introduced by the viscous damping matrix $\mathbf{C}$, which is constructed in the classic Rayleigh damping format. The force vector $\mathbf{f}$ consists of external force components at each degree of freedom (DOF) for each time step and the vector $\mathbf{r}$ contains the corresponding degrees of freedom of the structure. Note that the components in the stiffness matrix $\mathbf{K}$ dependent on the deflection $\mathbf{r}$ of the structure. The response of the structure is calculated by the Newmark direct integration method. The Newton-Raphson method is used to achieve force equilibrium in each time step and the update of the stiffness matrix follows the procedures described in [14]. In order for the equilibrium algorithm to converge and remain stable a time step increment of $\Delta t = 0.01\,\mathrm{s}$ is used in the numerical simulation. This increment is sufficient to resolve the governing dynamics of the numerical models considered in Section 4. Finally, equilibrium within the Newton-Raphson iteration loop is accepted when the force and displacement residuals are below 10^{-4} relative to the corresponding load and displacement increments, respectively.

3.2. *Simulation*

The ANN can in principal be designed to simulate any desired output from the FEM analysis. For simplicity the networks used in the following examples are trained to predict the horizontal deflection of the risers. Note that the ANN output only gives the response at one desired location and hence, as oppose to the FEM model, not the response of all model DOFs. The horizontal platform motion together with the riser response r of previous time steps constitute ANN input vector

$$\mathbf{x}_t = [u_t \quad u_{t-h} \ldots \; u_{t-dh} \quad r_{t-h} \quad r_{t-2h} \ldots \; r_{t-dh}]^T, \tag{7}$$

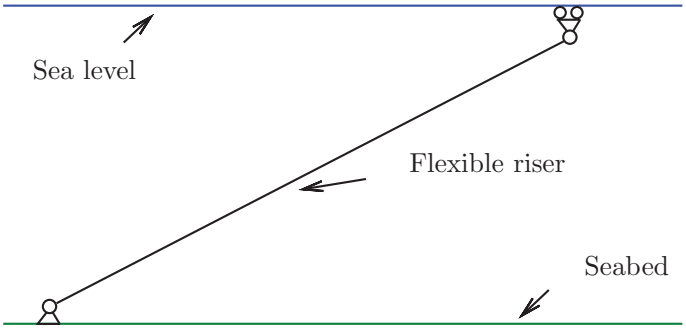


Figure 3. Sketch of riser in initial configuration.

where u is the prescribed platform motion, r is the horizontal riser response of a selected node in the FEM model, t denotes current time, h is the time increment and d is number of previous time steps included in the input, i.e. the model memory. Note that the ANN not necessarily uses the same time step increment as the numerical integration algorithm, why h may be different from Δt introduced in Section 3.1. The ANN output of the simulations is the current time horizontal deflection of the selected node in the riser model,

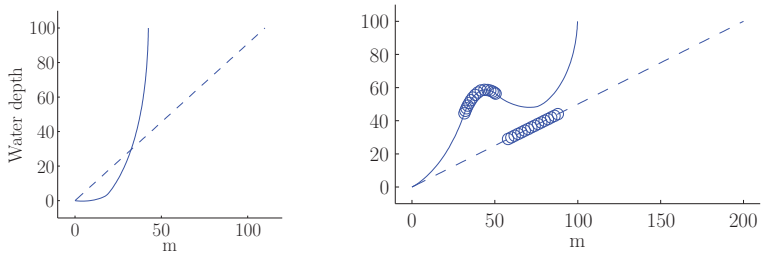
$$y_t = [r_t] \tag{8}$$

The output layer in an ANN can in principal have any number of elements. However, since the ANN used in this example simulates the response of a single node only, the output y in (8) is simply a scalar, and not a full vector as in (1).

After training the ANN is able to predict the riser response to a similar platform motion history as the one used to generate the training data. The ANN training, i.e. the minimization of the error function (2), is performed with a certain representative training data set. It has been shown that an ANN with one hidden layer and a sufficient number of elements in the hidden layer can replicate any continuous function [15]. However, the purpose of the ANN is not to replicate a specific set of data, but to learn the underlaying characteristics of the structural FEM model. This means that the ANN must be tested during training. For each iteration the test result must be stored in order to keep track of the development in the validation error to avoid overfitting with respect to the specific data in the training set. As the ANN validation is performed using 'fresh' data all the data generated by the FEM model must be divided into a training part and a validation part. As mentioned, the validation error E_{test} is calculated in the same way as the training error in E_{train} (2) but with respect to fresh data, which has not been part of the network training.

4. Two riser configurations

The sequential hybrid procedure is demonstrated for two different riser configurations: A steel catenary riser (SCR) and a Lazy-wave riser. Both risers are installed at 100 m water depth, as sketched in Figure 2. The demonstration is divided into three main steps. The first two steps are conducted by the FEM model. The purpose of the first task (step 1) is to get the riser model in place, i.e. in equilibrium. For both riser configurations this first step is initiated from a situation, where the riser is placed in a straight line between the sea bottom and the sea level. The riser



(a) SCR configuration. (b) Lazy-wave configuration.

Figure 4. ——— Final configuration, – – – Initial configuration; 'o' Buoyancy modules.

is prevented against translations by the simple support at the bottom, while at the sea level it is restrained against vertical displacements. The initial configuration is shown in Figure 3. Since the numerical FEM model does not consider hydrodynamic damping and any added mass effects, the mechanical and model properties are both adjusted so that they give realistic dynamic behavior. In the present case this specifically involves adding significant supplemental mass to the numerical riser models, simply to secure the correct order of magnitude of the natural frequencies. Hence, the FEM model does not explicitly represent an entirely realistic flexible riser, but is merely to be considered as a numerical generator of useful data for the demonstration of the hybrid method with ANNs. For the lazy-wave configuration a lifting force, corresponding to the lift from the intermediate buoyancy modules, are applied to a sequence of adjacent element nodes in the FEM model. The placement of the riser is now carried out as a (quasi-static) dynamic simulation, where the top node is initially released, whereafter the forces from gravity and buoyancy place the risers in their representative equilibrium states. The initial and the final configurations for the SCR and lazy-wave riser are shown in figures 4(a) and 4(b), respectively.

Having the risers in the desired final configuration the next step (step 2) is then to apply a horizontal motion history to the top node. This motion history corresponds to the horizontal motion of a floating offshore platform with a mean (offset) of 2.6 m, a standard deviation of 2.4 m, a dominant wave frequency of about 0.1 Hz and a low frequency motion with a period of around 80 s.

This study considers the simplest possible example of riser structures. The numerical model is two dimensional, there are no forces from waves and current acting on the riser and there is no interaction with the seabed, which means that the risers are hanging freely. The only external effect on the structure is the prescribed horizontal motion (surge) of the top node at the sea level. Even though hydrodynamic effects and vertical motion (heave) at the top node of the riser contribute significantly to the horizontal riser deflection, in order to keep the analysis as simple as possible, these effects are omitted in following. The FEM analysis calculates the time histories for all DOFs at all element nodes of the entire FEM model. However, in this numerical illustration of the sequential hybrid method we are merely interested in the horizontal motion of a selected number of element nodes along the riser. As mentioned earlier, the top tension forces in risers and mooring lines can fairly easily be predicted by properly trained neural networks, [8, 16]. But in order for this hybrid method to become useful in riser design we need a way to make it work for all critical locations along the risers, for example the touch down zone at the seabed. Otherwise, the computationally expensive simulations based on equilibrium iterations must be conducted anyway. However, as this paper merely

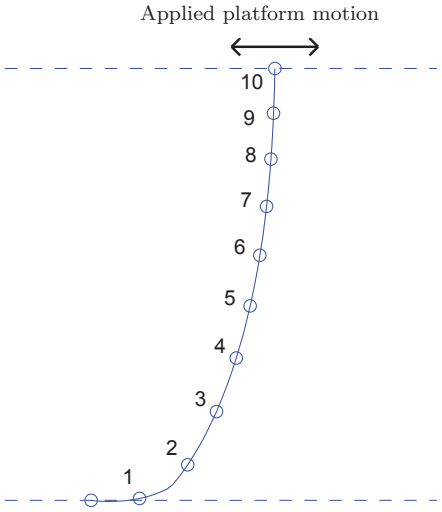


Figure 5. Sketch of FEM model.

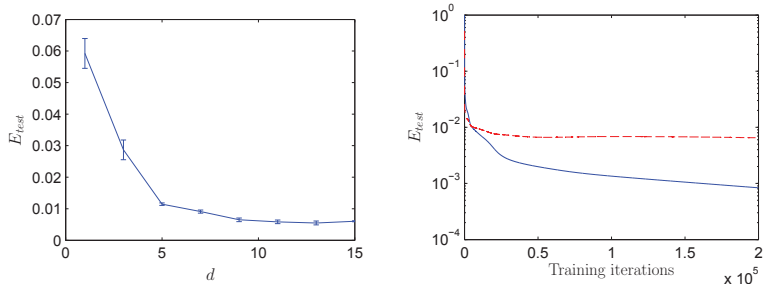
aims to illustrate the sequential ANN method on a very simple FEM model, the response simulations do not consider specific fatigue critical hot spots on the riser structures, but only serve to demonstrate that all locations from top to bottom may be covered by the present ANN simulation technique.

The last step (step 3) is to set up and train an ANN with respect to the data generated in the FEM analysis (step 2). The ultimate goal for this method is of course to predict pipe section forces at the hot spots of the analysis, for example the tension forces in the touch down zone directly using only the platform motion at the sea level as input to the ANN. But for simplicity this first demonstration in this paper considers the horizontal motion at a number of selected and regularly spaced element nodes. A sketch of the FEM model and the selected nodes for the SCR configuration are shown in Figure 5.

4.1. *Steel catenary riser*

As mentioned previously, the architecture of the ANN depends on a number of network variables. Because there are no explicit rules for determining the optimal ANN, determining an appropriate network architecture is often done by trail and error measures. Using this approach for the SCR data shows that a hidden layer with 35 neurons gives accurate simulations, while still having a relatively compact and efficient ANN. In Figure 6(a) the test error determined by (2) is plotted with respect to model memory, which is the number of previous time steps that are used as network input - corresponding to d in (7). The figure shows that including memory in the model reduces the error significantly. On the other hand it is seen that including more than approximately 10 steps does not seem to improve the ANN performance noticeably. Thus, in the following $d = 10$ is chosen.

Earlier studies have shown that the size of the time step in the ANN simulation can be significantly larger than the corresponding time step used in the FEM analysis, see [16]. Thus, in this example a time step size of $h = 2$ s has proven to be sufficient. This corresponds to approximately five individual discrete data points for each wave period. By omitting the majority of the FEM data the time spend on ANN training is significantly reduced and it furthermore also improves the performance of the ANN simulations. The training and the validation are now



(a) ANN memory, d.

(b) Error reduction during ANN training.

Figure 6. Finding optimal ANN architecture.

both carried out using 1200 s (20 minutes) of response history generated by the FEM model. The full time history is divided into 700 s of training data and 500 s of validation data, with a time step size of 2 s. This leads to 300 discrete data points for both the training set and the corresponding validation set.

Now that the ANN architecture has been chosen the next step is to determine for how large a spatial separation between the network input and output variables the ANN can obtain accurate simulation results by just using the prescribed motion of the riser top node as input. Figure 6(b) shows the development in ANN error during training. The training error E_{train}, which is minimized by the training algorithm, is represented by the solid curve, while the dashed curve shows the ANN validation error E_{test}. This latter measure indicates how well the ANN performs with respect to fresh data. Since the goal of the procedure is to train the ANN to actually simulate the structural response with respect to unknown data, the implicit objective of the training is therefore to minimize E_{test}. Hence, it is important to keep track of the development in the validation error during the training to avoid overfitting with respect to the specific training data. An increase in E_{test} often indicates overfitting, in which case the training should be terminated. Figure 6(b) shows how the error measure is reduced during training for the ANN simulating the horizontal motion of node 2 with the prescribed platform motion motion at node 10. Both training error (solid curve) and validation error (dashed curve) are plotted with respect to the number of iterations in the training procedure. It is seen that both error measures are reduced as intended.

Figure 7 summarizes the sequence of simulations performed by the sequential hybrid method for the SCR. Figure 7(a) shows that the simulation for the SCR is accurate all the way down to node 2 when using just the top node (node 10) as ANN input. Figure 7(b) shows that going all the way to the bottom of the riser at node 1 the ANN with input from node 10 still captures the overall pattern, but the simulation results are not as accurate as for node 2. However, because good simulation accuracy is available at node 2 with respect to the top node motion, the simulation procedure for the bottom node 1 is now modified, so that the motion records of both node 10 and node 2 are used as input to the ANN when trying to simulate the response at the bottom node 1. This augmentation of the procedure is the basis of the sequential analysis, where ANN simulations not only depend on the original top node motion, but also on intermediate response histories obtained by the ANN. In the present case this sequential procedure obviously requires two ANNs. The training of the additional ANN is done using those data already generated by the FEM calculation. Hence, it is very important to stress that the augmentation of the hybrid ANN procedure does not require any addi-

Table 1. Validation error for the three simulations in Figure 7.

input/output:	10/2	10/1	10,2/1
E_{test}:	$6.5 \cdot 10^{-3}$	$10.0 \cdot 10^{-3}$	$9.4 \cdot 10^{-3}$

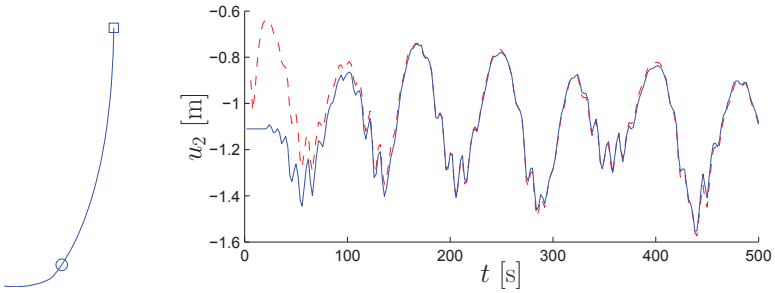
tional time consuming FEM analysis. The training itself is a computationally fairly cheap task, in particular when comparing the effort to the significant reduction in computational time provided by the use of the hybrid method. With that in mind the additional ANN is not considered as a significant degradation of the performance of the hybrid method, and cost associated with this network expansion is considered to be insignificant. The results of having two nodes as ANN input when trying to simulate the response at node 1 are shown in Figure 7(c), and it is found that using two input histories to the ANN yields a much more accurate simulation result.

One of the challenges with this sequential form of the hybrid method is to decide on an appropriate acceptance level for the validation error. In Table 1 the validation error is listed for the three networks that have been used to generate the curves in Figure 7. The error is calculated following $2 \cdot 10^5$ training iterations, and it is questionable whether it is possible to say anything conclusive based on these error measures. It would obviously be convenient to have a threshold level for the error to determine when an ANN output is sufficiently accurate. However, establishing such a threshold value would require more examples and experience with this method, and the present paper therefore only addresses the proof-of-concept. Most likely, some form of visual inspection will under any circumstances be part of the assessment of the ANN performance.

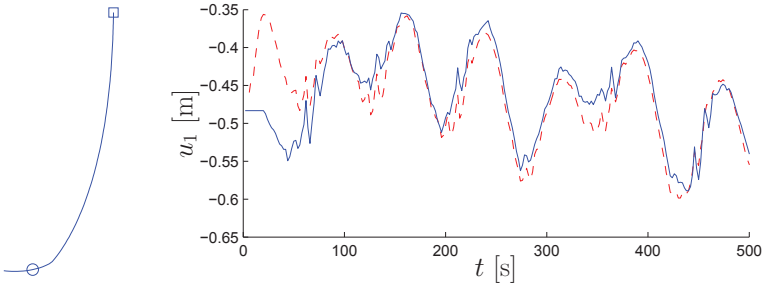
The above conclusions concerning the sequential procedure of the hybrid method are summarized in the ANN flowchart in Figure 8. The first network ANN_2 simulates the response u_2 of the SCR at node 2, and this response prediction is subsequently used as input for ANN_1, which computes the response u_1 of the bottom node 1. Together with the motion of the top node the output of ANN_2 is therefore used as input to the next network ANN_1, which is based on these two input signals and predicts the horizontal response of node 1.

4.2. *Lazy wave*

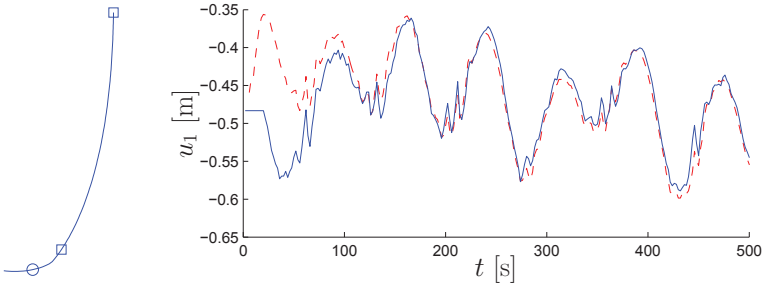
The sequential analysis in the previous section for the SCR is now repeated for the somewhat more complex lazy-wave riser configuration. All model parameters are the same as for the SCR model, i.e. same element stiffness, same number of elements and same water depth. The only difference between the models of the two riser configurations is the introduction of the intermediate buoyancy modules attached to the riser, as indicated in both figures 2 and 4. The ANN training and validation is again based on 20 minutes of simulation data. However, this divided into 750 s for training and 450 s for validation. The architecture of the ANN for the lazy-wave riser model is the same as the one used in the SCR simulations. Again a sequence of ANNs is used in this hybrid method strategy, where the output of a particular ANN is used as input to the next ANN, simulating the response of the adjacent node in the riser model. This sequential approach of course requires a fine balance between accuracy and the number of nodes along the riser used by the sequential hybrid method. For every time an ANN is used in the sequence of simulations an error is accumulated and passed on to the next ANN. Thus, even small inaccuracies in the intermediate simulations will be accumulated and potentially affect the accuracy of



(a) Response at Node 2 using Node 10 as input.



(b) Response at Node 1 using Node 10 as input.



(c) Response at Node 1 using Node 2 and 10 as input.

Figure 7. □: ANN input, ○: ANN output, ——— ANN, – – – FEM

the target response, e.g. at the bottom node of the riser. In addition to accumulation of error, an additional ANN also yields additional time spend on training, validation and simulation. Therefore, it is obviously desirable to keep the number of required ANNs in the sequential hybrid approach at a minimum, while still being able to generate accurate and robust results.

Another drawback of introducing additional neural networks in the sequence is that each network is associated with an initial transient time region, where the network must get synchronized with the input, which means that in a sequential procedure the removal of these transient parts for each network reduces the effective length of the corresponding time histories. In practice this means that the length of the time history of the top node at sea level increases with the number of ANNs required along the riser.

For the lazy wave riser configuration it turns out that, in order to keep the simulation error sufficiently low, it is necessary to have a distinct ANN for each of the selected nodes in Figure 9. Figures 10-12 show how the individual networks perform at different locations along the flexible lazy-wave riser. Except for the first ANN in Figure 10(a) all networks use simulated input generated by the previous

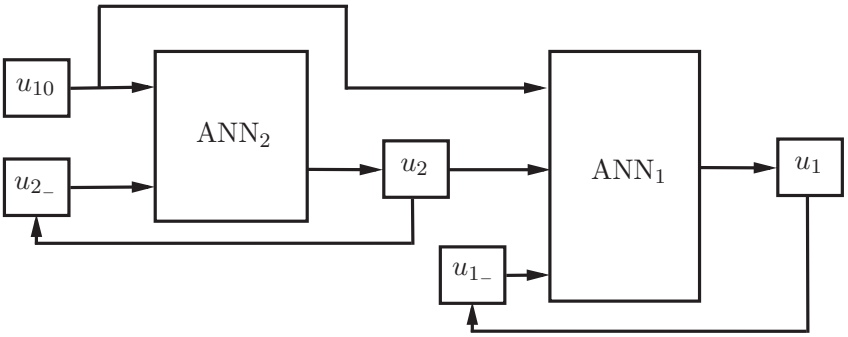


Figure 8. ANN structure for SCR simulation.

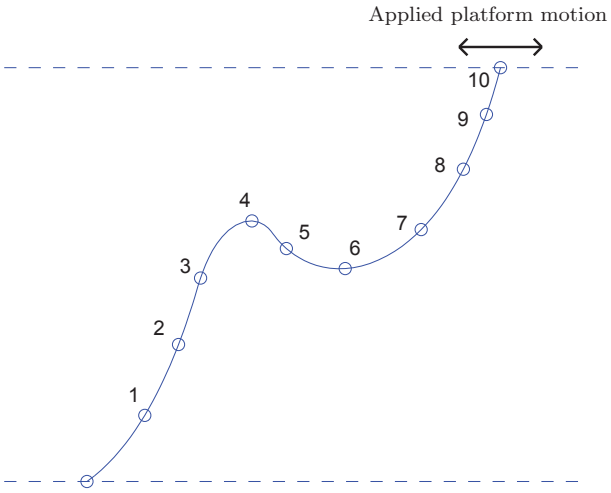
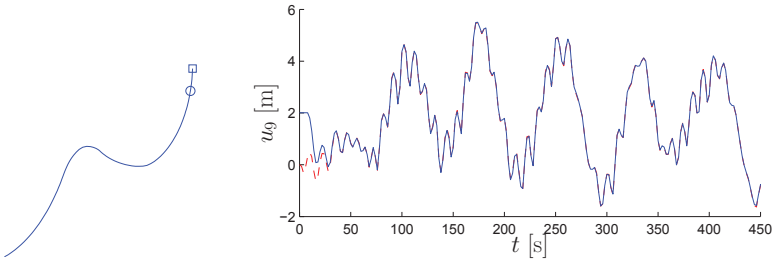


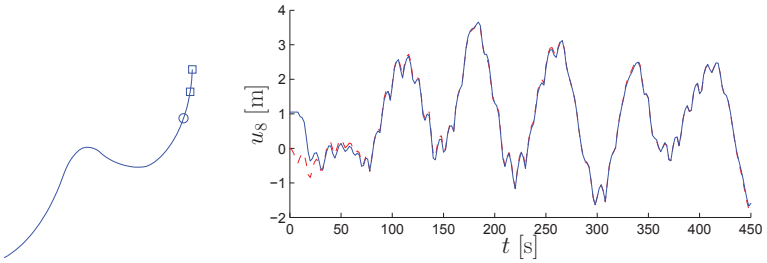
Figure 9. Sketch of FEM model.

ANN, as illustrated in figures 10-12 where the boxes on the riser sketches indicate which nodes that are used as ANN input while the circles indicate the ANN output node. This means that any previous inaccuracies are somehow passed on and thereby remain in subsequent simulations. This phenomenon can be detected in the series of simulations shown in figures 10-12. For example, in Figure 11(a) a small simulation inaccuracy seemingly occurs after approximately 350 s. This error is not pronounced in the previous simulations shown in Figure 10. However, when inspecting the subsequent simulations in Figure 11 it is seen that this validation error still exists. Despite the accumulation of errors, the sequential hybrid method still manages to maintain a very high degree of accuracy in the response simulations all the way along the riser down to the final node 1 at the sea bed.

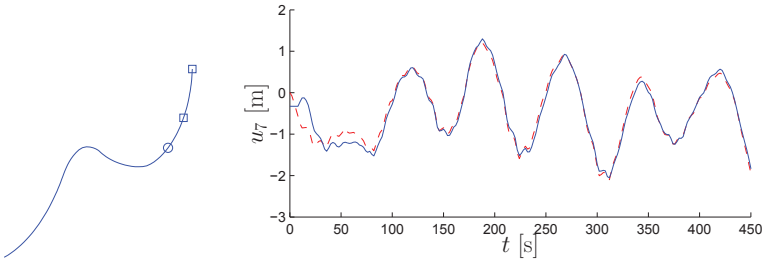
Figure 13 shows the ANN flowchart used for the lazy-wave riser configuration in the case where the horizontal motion of node 1 depends on the response obtained by the previous nine ANNs of the sequential analysis. The actual time spend on simulating after having designed and trained is insignificant, compared to the time used to perform the original FEM simulation. As for the SCR, the training of the nine ANNs can be done entirely by the data generated for a single FEM analysis.



(a) Response at Node 9 using Node 10 as input.



(b) Response at Node 8 using Node 9 and 10 as input.



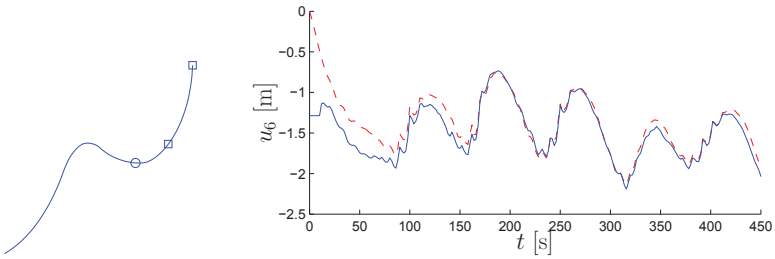
(c) Response at Node 7 using Node 8 and 10 as input.

Figure 10. □: ANN input, ○: ANN output, ——— ANN, – – – FEM

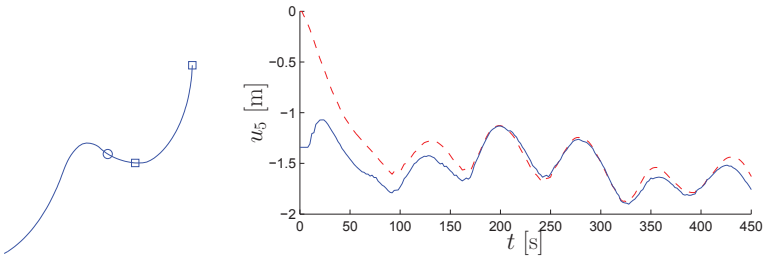
5. Concluding remarks

For the sequential hybrid method to be beneficial the extra time spend on setting up and training the neural networks must be less than the time spend on calculating the required amount of time domain simulation when using the FEM model. Time domain simulations using classical numerical methods can in many cases take hours or even days before a sufficient amount of data is obtained, whereas ANN simulation, once the series of networks is trained and ready, is so fast that simulation time is usually a matter of seconds. Figure 14 illustrates a reasonable relationship between required simulation time and computation time for the two methods.

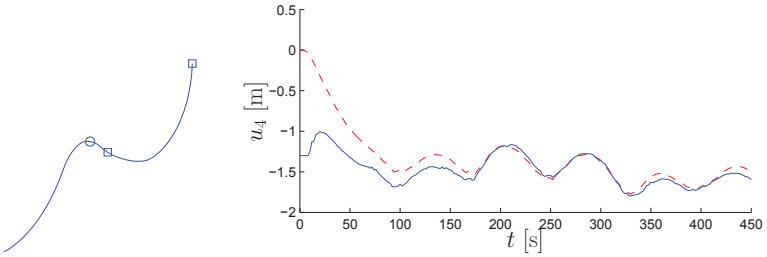
In the example presented in this paper the difference in computational time spend on the individual simulations by using the ANN and using the classic FEM model lies around a factor of 100 - that is once the ANN is constructed, trained and validated. This factor obviously depends on the type of structure, the loading conditions, the model configuration and various other key elements. However, the examples in this paper indicate that the sequential hybrid method holds a great potential. Furthermore, the reduction in calculation cost may be increased even more if it is possible to avoid one or more post processing steps so that, instead of just predicting the structural response, the ANN can in fact be trained to predict material stresses or structural damage directly based on the FEM model input



(a) Response at Node 6 using Node 7 and 10 as input.



(b) Response at Node 5 using Node 6 and 10 as input.



(c) Response at Node 4 using Node 5 and 10 as input.

Figure 11. □: ANN input, ○: ANN output, ——— ANN, – – – FEM

data. In this paper FEM and ANN simulations have been compared only by visual inspection of the relatively short time domain simulations in Figure 7 and figures 10-12. It would be interesting to evaluate the ANN accuracy by comparing statistics on long time domain simulations. This will be part of future work. Another obvious next step will be to test the sequential method on more detailed numerical riser models, e.g. on data generated by recognized commercial software such as RIFLEX or OrcaFlex.

References

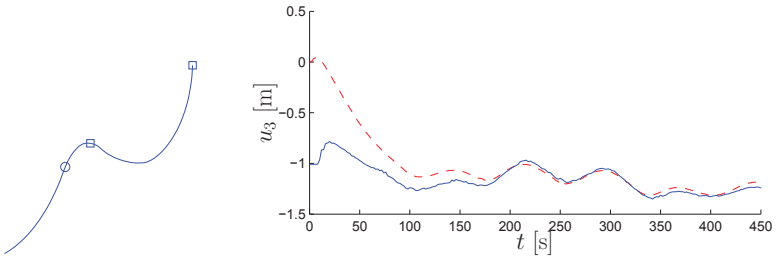
[1] DNV. *Offshore Standard DNV-OS-F201 - Dynamic Risers*. Det Norske Veritas, October 2010.

[2] DNV. *Recommended Practice DNV-RP-F204 - Riser Fatigue*. Det Norske Veritas, July 2005.

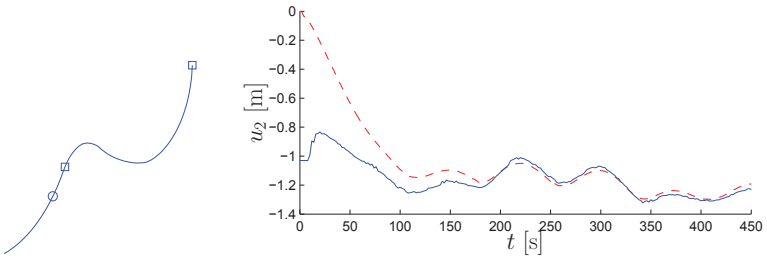
[3] L.V.S. Sagrilo, Z. Gao, A. Naess, and E.C.P. Lima. A streightforward approach for using single time domain simulations to assess charavteristic response. *Ocean Engineering*, 38(5):1464–1471, August 2011.

[4] H. Adeli. Neural networks in civil engineering: 1989-2000. *Computer-Aided Civil and Infrastructure Engineering*, 16:126–142, 2001.

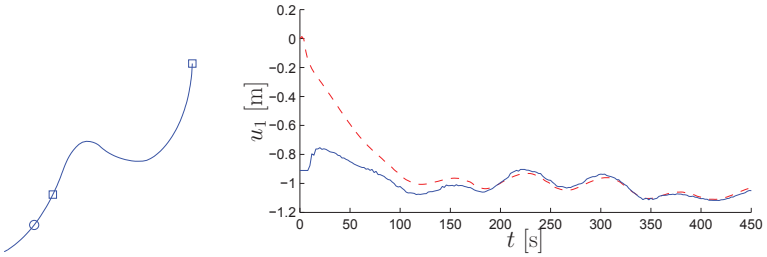
[5] S. Gupta, A. Ray, and E. Keller. Symbolic time series analysis of ultrasonic data for early detection of fatigue damage. *Mechanical Systems and Signal Processing*, 21:



(a) Response at Node 3 using Node 4 and 10 as input.



(b) Response at Node 2 using Node 3 and 10 as input.



(c) Response at Node 1 using Node 2 and 10 as input.

Figure 12. □: ANN input, ○: ANN output, ——— ANN, – – – FEM

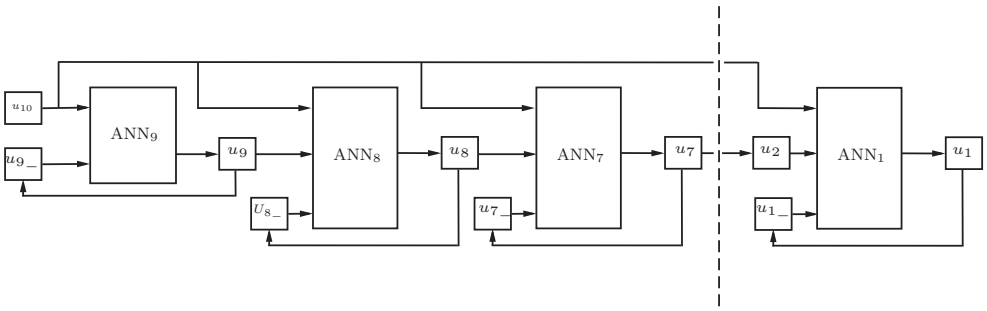


Figure 13. ANN structure for Lazy-wave simulation.

866–884, 2005.

[6] J. Klemenc and M. Fajdiga. A neural network approach to the simulation of load histories by considering the influence of a sequence of load cycles. *International Journal of Fatigue*, 24:1109–1125, 2002.

[7] K. Ordaz-Hernandez, X. Fisher, and F. Bennis. Modal reduction technique for mechanical behaviour modelling: Efficiency criteria and validity domain assessment. *Journal of Mechanical Engineering Science*, 222:493–505, 2007.

[8] R. Guarize, N.A.F. Matos, L.V.S. Sagrilo, and E.C.P. Lima. Neural networks in dynamic response analysis of slender marine structures. *Applied Ocean Research*, 29: 191–198, 2008.

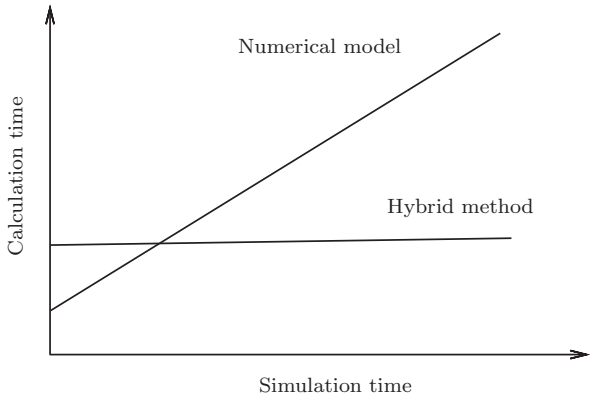


Figure 14. Potential benefit of the hybrid method when long time domain simulations are required.

[9] M. Yazdchi and M.A. Crisfield. Non-linear dynamic behaviour of flexible marine pipes and risers. *International Journal for Numerical Methods in Engineering*, 54: 1265–1308, 2002.

[10] B. Warner and M. Misra. Understanding neural networks as statistical tools. *The American Statistician*, 50:4:284–293, 1996.

[11] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006.

[12] C. M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1995.

[13] R. Fletcher. *Practical Methods of Opimization*. Wiley, 1987.

[14] S. Krenk. *Non-linear Modeling and Analysis of Solids and Structures*. Cambridge University Press, August 2009.

[15] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303–314, 1989.

[16] N.H. Christiansen, P.E.T. Voie, N. Sdahl, and J. Hgsberg. Efficient mooring line fatigue analysis using a hybrid method time domain simulation scheme. *Procedings of the OMAE2013 International Conference on Ocean, Offshore and Arctic Engineering*, page TBA, 2013.

DTU Mechanical Engineering
Section of Solid Mechanics
Technical University of Denmark

Nils Koppels Allé, Bld. 404
DK- 2800 Kgs. Lyngby
Denmark
Phone (+45) 4525 4250
Fax (+45) 4593 1475
www.mek.dtu.dk
ISBN: 978-87-7475-410-7

DCAMM
Danish Center for Applied Mathematics and Mechanics

Nils Koppels Allé, Bld. 404
DK-2800 Kgs. Lyngby
Denmark
Phone (+45) 4525 4250
Fax (+45) 4593 1475
www.dcamm.dk
ISSN: 0903-1685